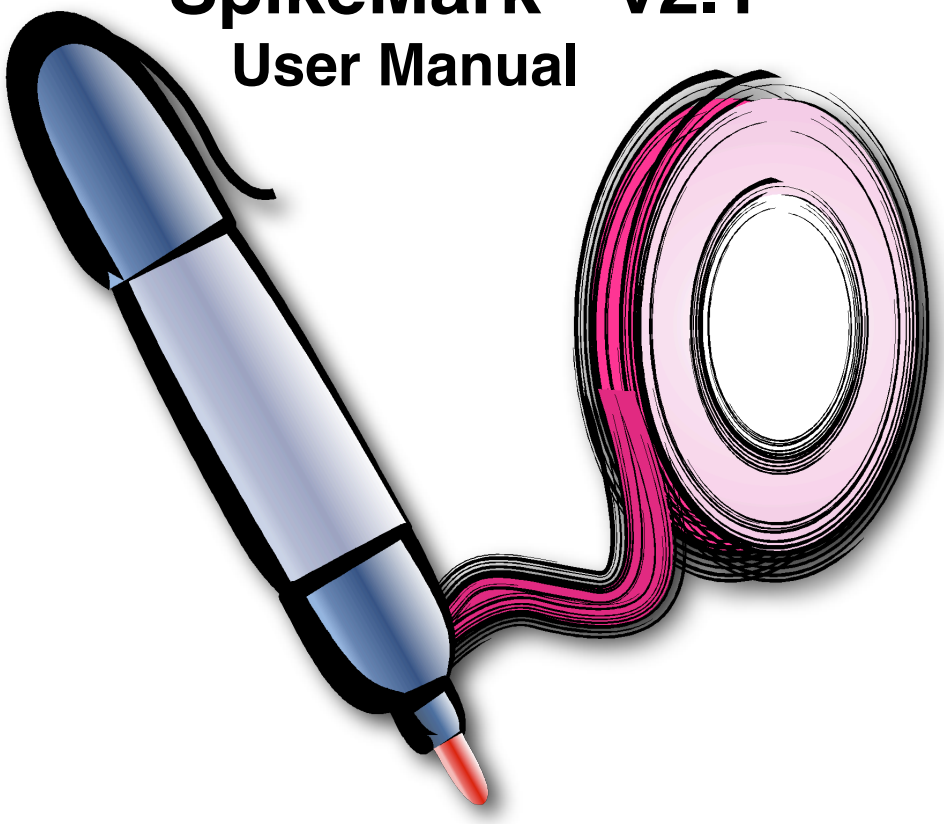
SpikeMark™ v2.1
User Manual



Welcome To SpikeMark 5

A word about safety 5
System Requirements 5
Installing SpikeMark 6
Online tutorials 6

What's New Since Avista 7

Time-based cues 7
Spikes 7
Streamlined interface 7
3d stage view 7
Reset position 8
Target tolerance mode 8
Stagehand FX support 8

Quick Start 9

Physical connections 9
Setup a network connection 9
Confirm encoder feedback 15
Set the scaling factor 16
Set soft limits 17
Set default speed parameters 18
Write two cues 18
Run the cues 20
Basic tuning 22

Interface Overview 23

Create a New Show 24

Adding Stagehands 24

Adding motors 24
Adding Stagehand FX's 25

Motor Configuration 25

Renaming a Motor 26
Motor Status 27
Jogging a Motor Manually 29

Scaling Motor Position 29
Setting Software Limits For A Motor 30
Reset Motor Position 31
Motor Tuning 32
Max Position Error 35
Motor Cue Behavior 35
Motor Refresh Speed 36
Stagehand FX Configuration 37
Renaming an FX 38
FX Status 39
Manually Activating FX Outputs 40
Network Connections 41
Editing Stagehand Network Settings 41
Connecting to a Stagehand 42
Disconnecting from a Stagehand 42
Writing Cues 44
Adding Cues 44
Adding Motor Movements To A Cue 45
Change Motor Position In A Cue 47
Change A Spike Position 47
Change Motor Speed & Time In A Cue 49
Changing Cue Time 50
Adding an FX Movement 51
Deleting a movement 52
Deleting a cue 53
Linking Cues 54
Linking by completion 55
Linking by time 56
Linking by motor position 57
Linking by FX input 58
Delete a Link 58
Using the Stage Model 59
Navigating the Model 59

Setting the Stage Dimensions 60
Make a Winch Model 61
Make a Lift Model 63
Make a Curtain Model 66
Make a Turntable Model 69
Writing Cues in the Stage Model 72
Running Shows 80
Setting up the Workspace 80
Loading a Cue 82
Running a Cue 84
Deciphering Stagehand Status During the Show 88
Restoring a Cue - New in version 2.1! 89
Troubleshooting 92
Contact Us 92
The Stagehand network connection doesn't work 92
A motor didn't start to move when I ran a cue 92
SpikeMark showed a strange error message then I had to quit 93
I can't find the Log file to email it to support 93
A motor starts to run in a cue but stops abruptly before getting to spike 93
The motor runs full speed in the wrong direction 94
Appendix A - Basic Motion Control Concepts 94
Power 94
Position Feedback 94
Motor Tuning 95
Over-travel Limits 95

Welcome To SpikeMark

Welcome to SpikeMark, the software designed for theatrical automation. SpikeMark is built to make moving scenery easy and intuitive. As today's shows grow increasingly complex, SpikeMark offers a simple solution to bring the excitement of motorized scenery to every production. Using a Windows® PC and standard Ethernet hardware, you can have a motion control system that is easy to set-up and expand while being incredibly economical. Whether you need to move a deck winch, turntable, roll drop, or any other effect, SpikeMark allows you to focus on the custom aspect of your production by providing a standard way to control motion.

A word about safety

The spectacle created by motorized scenery is fantastic. Motion control allows huge scene changes to operate smoothly in a matter of seconds. But this power deserves the utmost caution. Unlike operating a computerized lighting console, or sound system, an automation system can pose a serious danger to performers and technicians. Large pieces of scenery moving at high speed can easily injure, even fatally, any person that is unwittingly caught in its path. Therefore, a large responsibility is placed on you, as the automation operator, to be alert and aware of the people that may be in harm's way. The **GO** button demands the same respect as the trigger on a loaded gun.

SpikeMark™ assists the operator in maintaining safety by providing constant information of every motor's position and status. A hard-wired emergency stop system, ShowStopper, provides an immediate way for the operator to shut down all the motion on the stage. Ultimately, it is the operator's judgment and ability to see any potential hazards that will keep the show safe.

System Requirements

To effectively use SpikeMark™ you will need:

1. A Windows® compatible PC (1.2 GHz or faster)
2. A dedicated graphics processor (recommended).
3. Windows® XP SP2 or better (Windows® Vista recommended).
4. Microsoft's .Net 3.5 framework
5. 10MB of free storage space
6. A network interface card (or built-in networking)
7. An Ethernet hub
8. Ethernet cables (1 for every device on the network)
9. One or more Stagehand™ motor controllers with electric motors (winches, turntables, etc.)
10. A ShowStopper™ emergency stop base station

Installing SpikeMark

To install SpikeMark™:

1. Insert the disc into your drive.
2. Navigate to the disc through Windows Explorer.
3. Launch SpikeMarkSetup.msi
4. Follow the onscreen instructions. If Microsoft's .Net 3.5 framework is not installed on your system, the installer will direct you to a Microsoft update page where the framework can be downloaded. Alternatively, you can download the framework via **Windows Update**.

Online tutorials

There are a selection of video tutorials on the Creative Conners website that cover all the basic uses of SpikeMark™. Visit http://www.creativeconners.com/prod_spikemark_tutorials.htm to view the tutorials.

What's New Since Avista

SpikeMark has a bunch of great new features, improving on Avista's legacy of being the easiest software to use for stage automation. We changed the name from Avista for this release to highlight one the handiest new features (and to avoid confusion with the latest operating system from a big company in Redmond, WA). Let's take a look at some at the new feature highlights.

Time-based cues

Cues can now be written using time parameters instead of speed and acceleration values. You can write cues by specifying how long the cue should run, and how long it should take for the motors to ramp up to speed. Allowing you to quickly write cues that match the timing of the show should help reduce tech-table stress.

Spikes

The Spikes feature is designed to mimic the usefulness real spike tape. Rather than trying to remember bizarre cue position numbers (like 251.13"), you can now assign names to commonly used positions ('on stage' = 251.13"). When writing cues, you can select a position from the motor's defined spikes.

Often during the rehearsal process, the spike positions need to be tweaked. If you were physically pushing the wagons around, you would just move the spike tape on the stage floor to a new position. Similarly, with the new Spikes feature, you simply edit the position of the spike in one place and all cues that use the spike are automatically updated, saving you from manually tracking changes through the show.

Streamlined interface

The entire interface has been re-worked to provide a more coherent presentation of SpikeMark's capabilities. Avista's floating windows have been replaced with a docked interface with resizable panes and some of the windows from Avista's interface have been merged together to make the interface easier to navigate.

3d stage view

The **Stage Schematic** has been replaced with the new **Stage Model** giving a three dimensional view of the theatre space. Wagons and turntables can be rotated along any axis making it easy to view lifts and deck winches in a more realistic fashion.

The new Spikes feature is integrated with the **Stage Model** to make it possible to build cues visually and accurately.

Reset position

A small, but often requested new feature is **Reset Position**. Motors can now be reset to an arbitrary position, rather than only being reset to zero. This is helpful if you need to swap out a Stagehand mid-show and need the new Stagehand to pickup the motor position exactly where the old Stagehand left off.

Target tolerance mode

Motors can optionally run their cues in **Target Tolerance Mode** which makes precise motor tuning less important. When run in this new mode, the motor will continue to try to reach target even if the tuning parameters are too sloppy to get to the target position in the precise time requested. You can enter a tolerance value that specifies when the motor should consider its position 'good enough' (1/4", 1/16", etc.).

Stagehand FX support

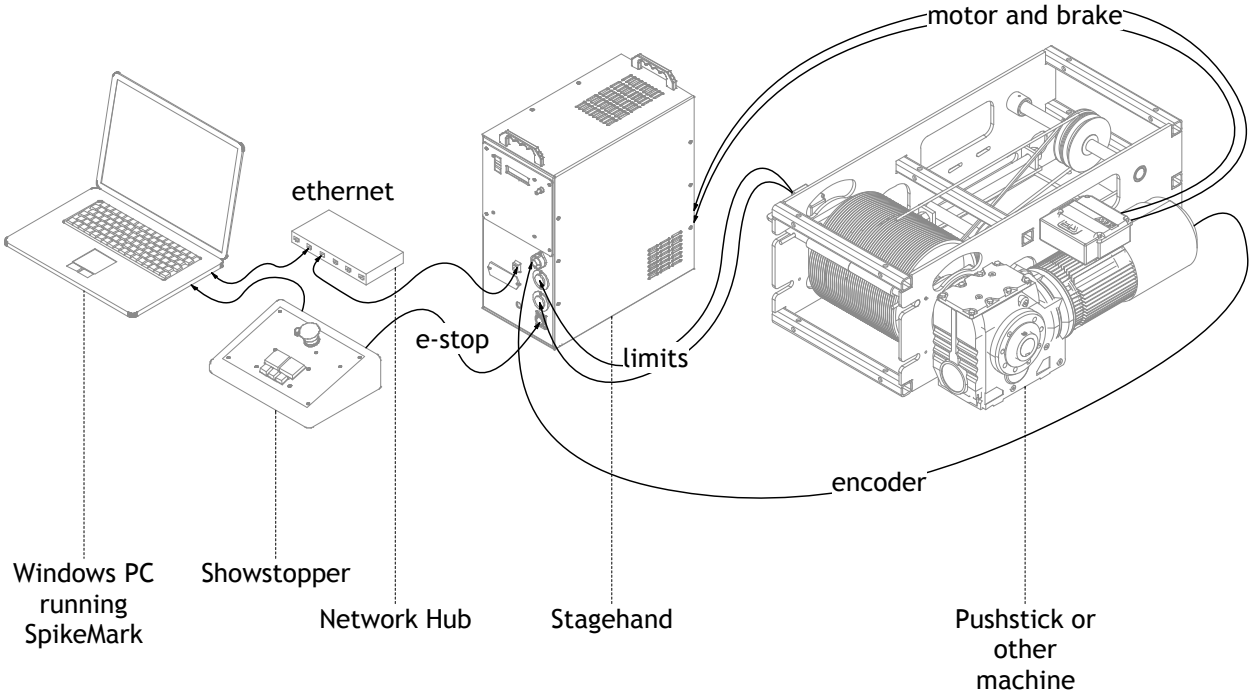
In addition to controlling motors, you can now control simple switched devices with the Stagehand FX and SpikeMark. When using the Stagehand FX with SpikeMark, each device gives you four switchable outputs and four readable inputs. Outputs can be used to turn simple devices on & off (lights, valves, small motors, etc.), inputs can be used as cue triggers for complex cue sequences.

Quick Start

The best way to learn to use SpikeMark, is to start using it. So let's jump in and get something moving!

Physical connections

Before anything will work, you need to make sure that all of your cables have been connected. Below is a diagram that shows how to plug everything in.



Setup a network connection

Once everything is plugged in, we need to test that the connections work. We'll start by confirming the Ethernet link works. SpikeMark and Stagehands communicate through static IP addressing. This means that your computer and all Stagehands need to have unique addresses. Much like a postman delivering letters to homes with address numbers nailed to the mailbox, SpikeMark sends it's commands to Stagehands with unique address numbers.

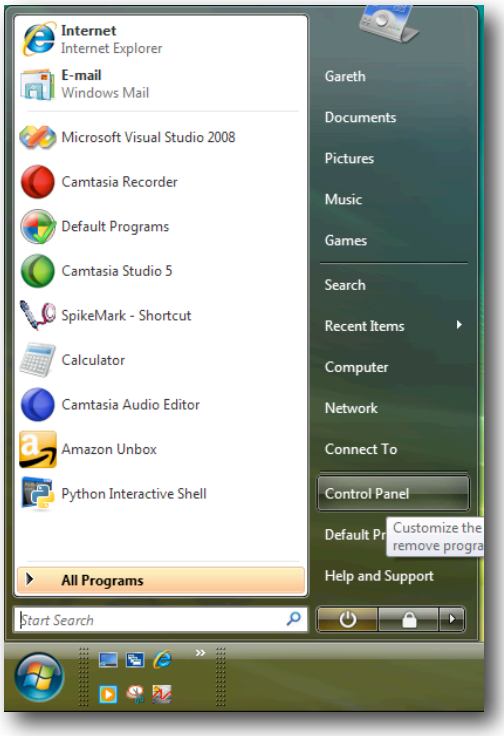
IP addresses are made up of four segments separated by a period (eg. "192.168.10.20"). The only trick to setting up a static IP network with SpikeMark and Stagehands is that the first three segments should always match. The last segment number needs to be unique.

These numbers should match. 192.168.10.20 This number should be unique.

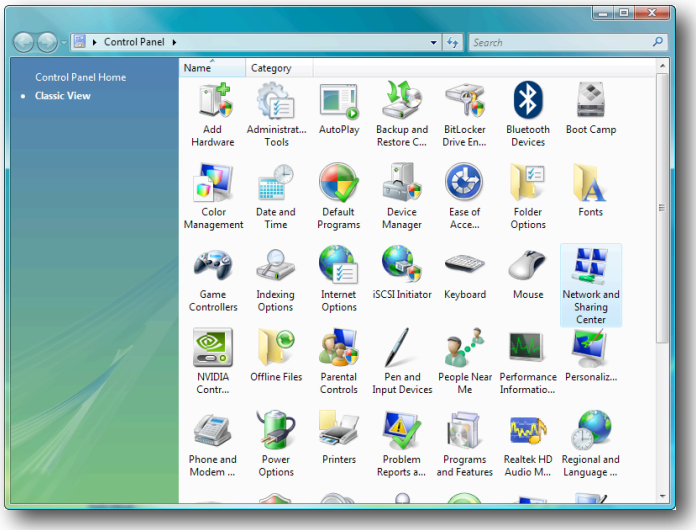
With that knowledge in hand, let's set the IP address of your computer.

In Windows Vista:

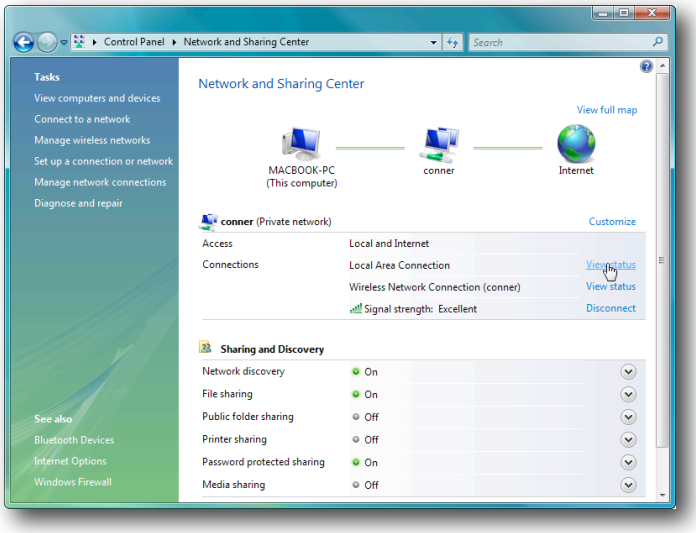
1. From the Start Menu Select Control Panel.



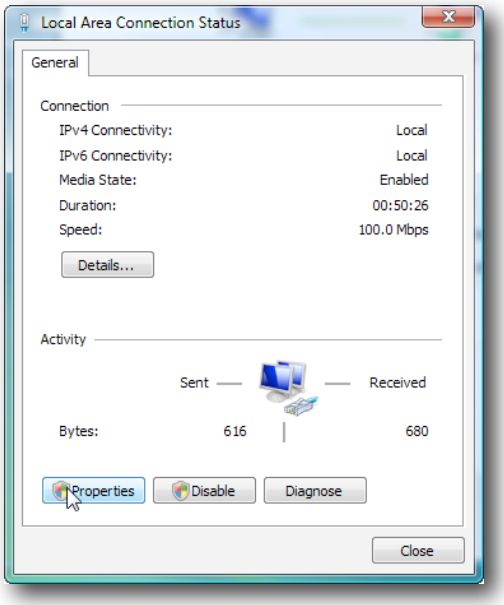
2. Double-click on Network and Sharing.



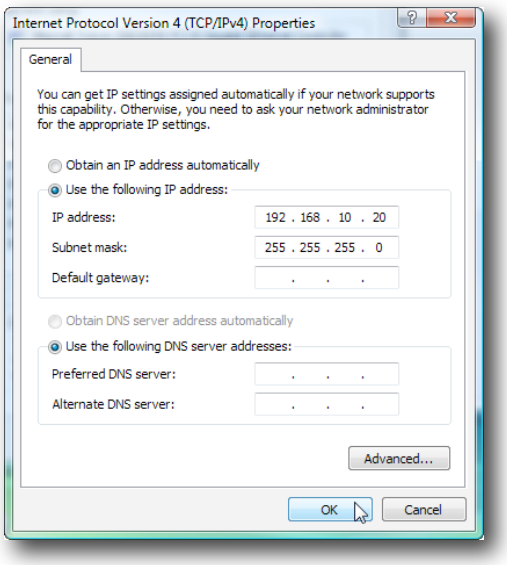
3. Click on the View Status link next to your Local Area Connection.



4. Select Internet Protocol Version 4 and click Properties.



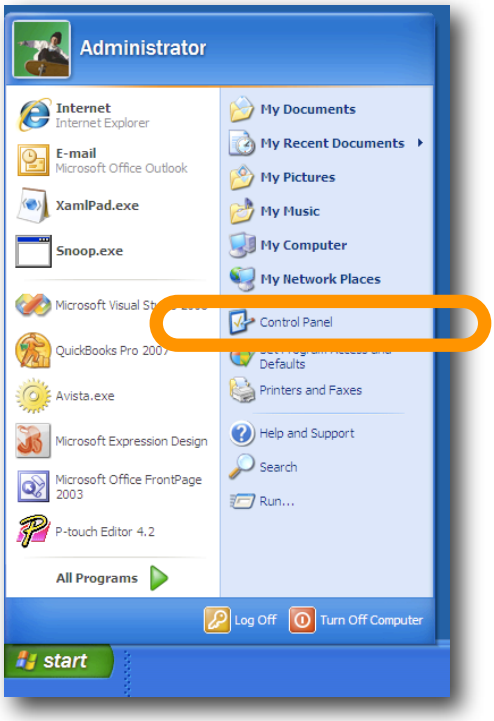
5. Select Use the following IP address and fill in the network properties as shown.


Internet Protocol Version 4 (TCP/IPv4) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address:
192 . 168 . 10 . 20
Subnet mask:
255 . 255 . 255 . 0
Default gateway:
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Advanced...
OK
Cancel

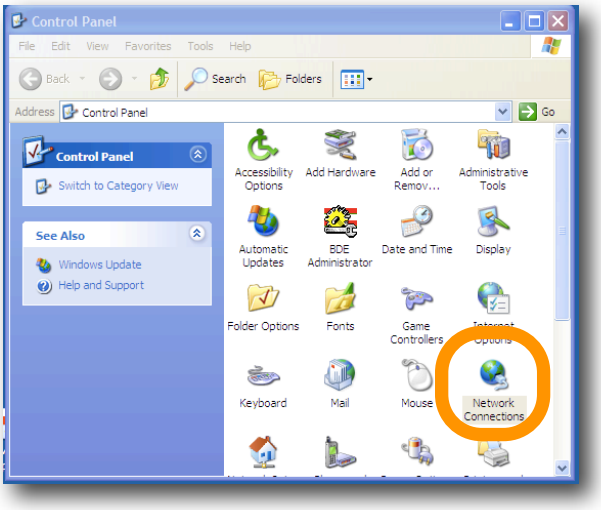
6. Click OK a bunch of times until you are back to your desktop.

In Windows XP:

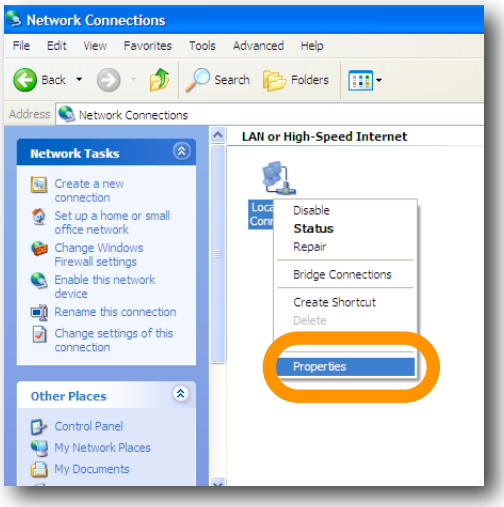
1. From the **Start Menu** select **Control Panel.**



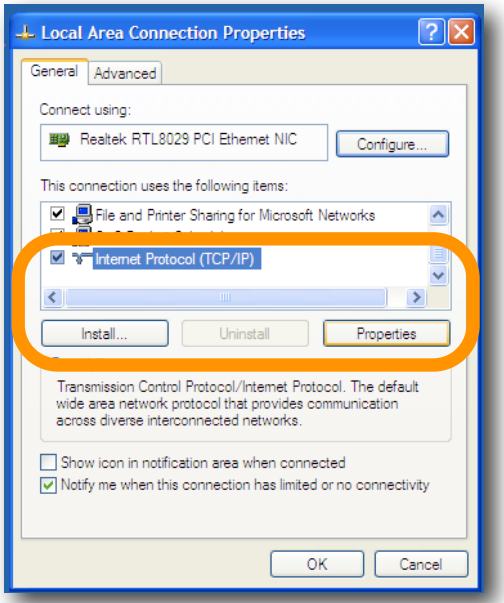
2. Double-click on the **Network Connections** icon.



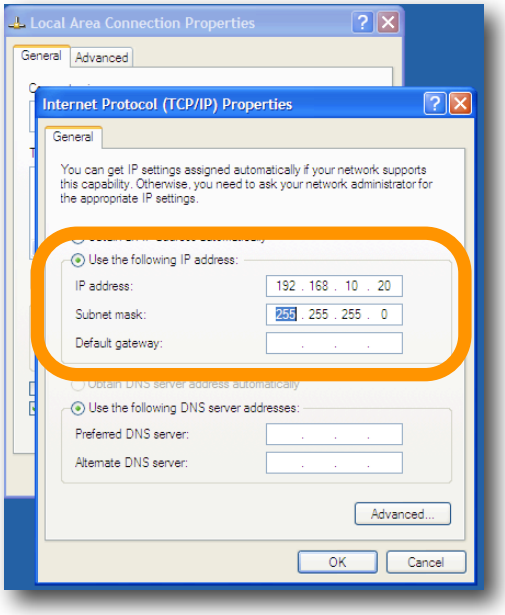
3. Right-click on the icon that represents your wired network connection (not a WiFi adapter).



4. Select **Internet Protocol (TCP/IP)** from the list and press the **Properties** button.


Local Area Connection Properties
General
Advanced
Connect using:
Realtek RTL8029 PCI Ethernet NIC
Configure...
This connection uses the following items:
File and Printer Sharing for Microsoft Networks
Internet Protocol (TCP/IP)
Install...
Uninstall
Properties
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.
Show icon in notification area when connected
Notify me when this connection has limited or no connectivity
OK
Cancel

5. Select **Use the following IP address**, and enter in the correct address.

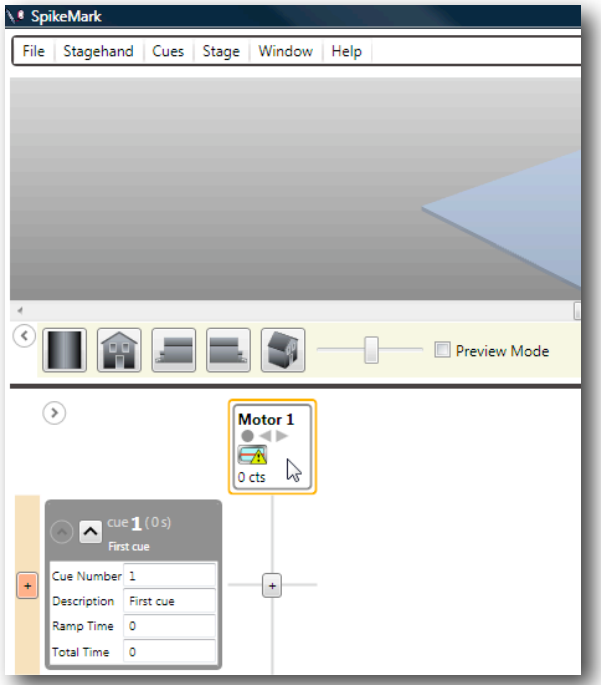

Local Area Connection Properties
General
Advanced
Internet Protocol (TCP/IP) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Use the following IP address:
IP address:
192 . 168 . 10 . 20
Subnet mask:
255 . 255 . 255 . 0
Default gateway:
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Advanced...
OK
Cancel

6. Press **OK** several times until your back to your desktop.

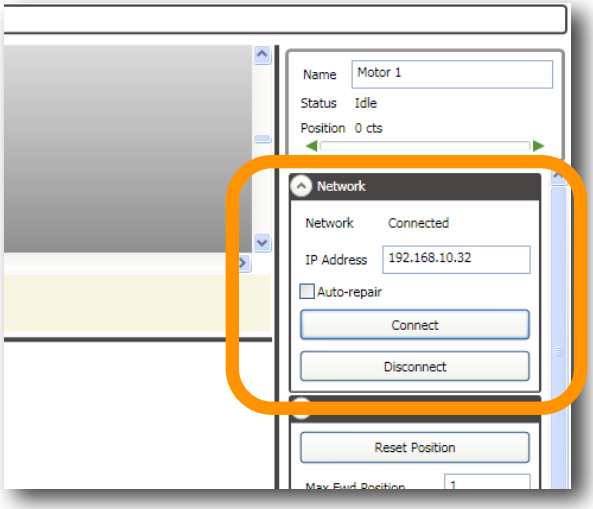
With the network address configured on your computer, we're ready to establish a connection to a Stagehand.

1. Launch SpikeMark.

2. Select the motor you want to connect by clicking on it in the **Cue Grid**. An orange highlight will appear around the selected motor.



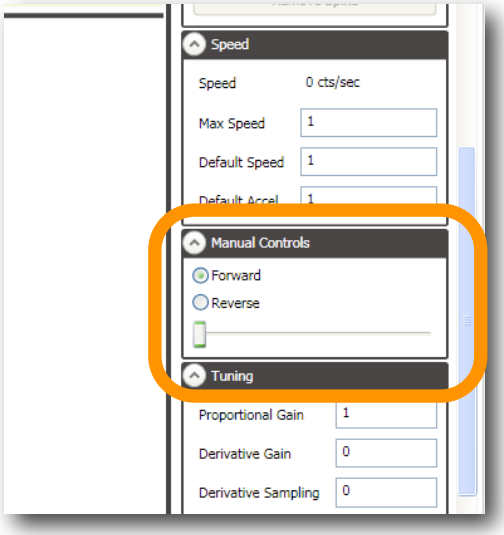
3. In the editing panel on the right side of the screen, enter the IP Address of your **Stagehand** and press **Connect**. Make sure you copy the IP Address exactly as it appears on the face panel of the **Stagehand** LCD display. The Network status should change from **Disconnected** to **Connected**. If there's a network error, check your IP settings on the computer and **Stagehand**, and then check your cabling.



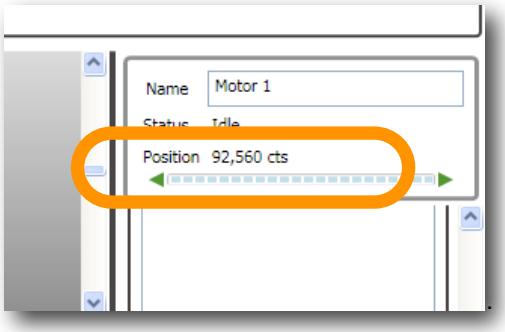
Confirm encoder feedback

1. Mark the motor's position on the stage with a piece of spike tape (this will be important in the next section).

2. With the motor still selected in the **Cue Grid**, scroll through the editing pane on the right side of the screen until you see the **Manual Controls** section.

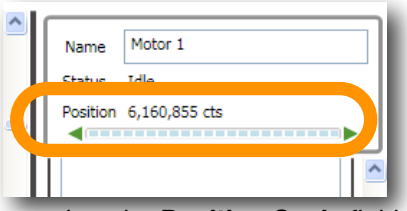


3. Select the **Forward** direction.
4. Make sure that there are no obstructions in the motor's path, and start slowly increasing the speed slider. You have to hold the mouse button down while the motor is moving, if you release the mouse button the motor will stop. It's a dead-man switch.
5. Confirm that the position number is increasing as the motor moves forward. The encoder and the motor have to match polarity for **SpikeMark** to work. So, if the encoder counts are decreasing while the motor is moving forward, you need to fix the polarity of the motor. To fix the polarity of the motor, just open up the motor plug and swap two of the armature wires (*DO NOT MOVE THE GROUND WIRE*)

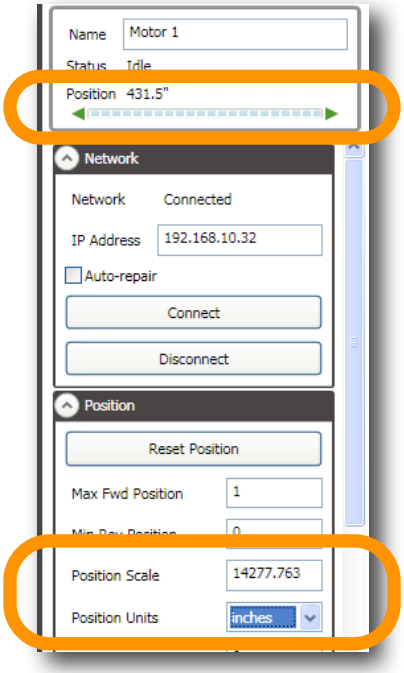


Set the scaling factor

1. Use the **Manual Controls** to move the motor as far forward as it can safely travel.
2. Measure the distance from the spike tape that you placed on the floor in Step 1 of the previous section to the current position of the motor.
3. Divide the **Position** by the number of inches traveled. For example 6,160,885 / 431.5" = 14,277.763 counts / inch.

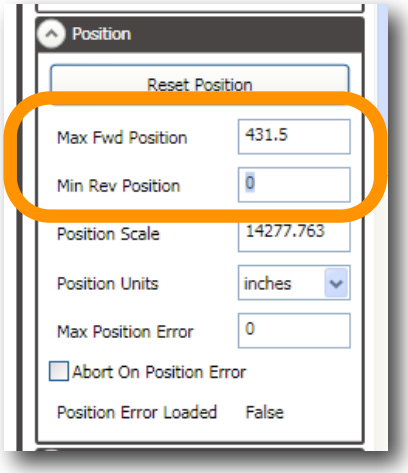


4. Type the computed scaling factor into the **Position Scale** field.
5. Select **inches** from the **Position Units** drop-down box. Note that the **Position** now displays 431.5”.



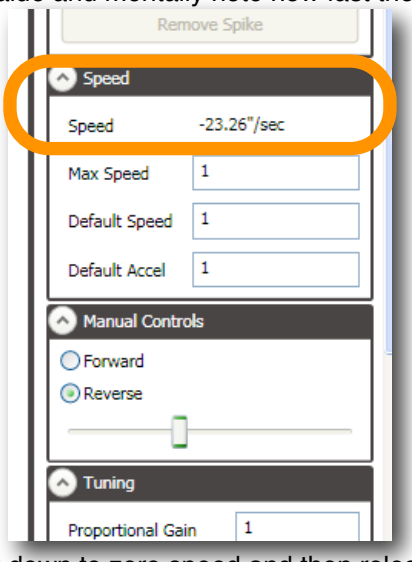
Set soft limits

1. Enter 0 for **Min Rev Position**. This will prevent you from accidentally writing cues less than 0.
2. Enter your current motor position as the **Max Fwd Position**. This will prevent you from writing cues greater than the current motor position. In this example, it would be 431.5”.

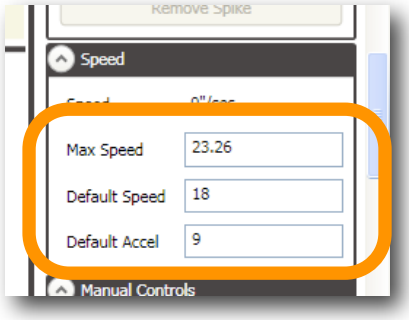


Set default speed parameters

1. Now let's figure out the motor's maximum speed. With the motor still selected in the **Cue Grid**, start moving it in reverse as fast as you feel comfortable using the **Manual Controls**. While moving, watch the **Speed** value and mentally note how fast the motor is moving.



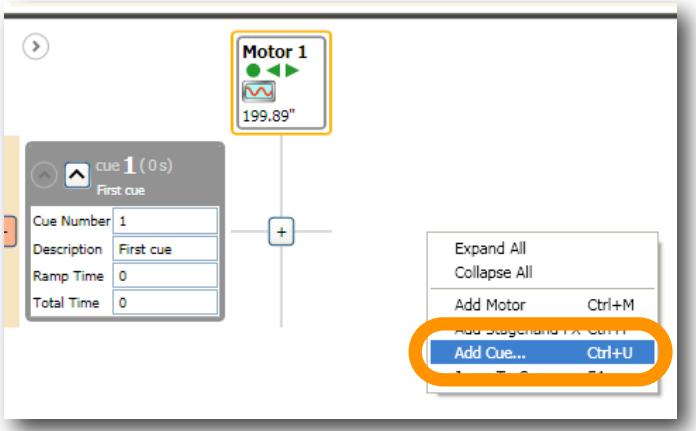
2. Slowly ramp the motor back down to zero speed and then release the mouse button to stop the motor.
3. Enter the speed value into the **Max Speed** box.
4. Enter a speed value into the **Default Speed** box. This number is the default speed that will be used when writing new cues. If you have a speed that is commonly used for the motor, using a **Default Speed** is a handy shortcut when writing cues. It's a convenience feature that can be overridden at anytime.
5. Enter an acceleration value into the **Default Accel** box. Acceleration values are given in inches/sec/sec (inches *per* second *per* second). I like to enter a value that is half of the **Default Speed**, which gives you a default acceleration that will last for 2 seconds.



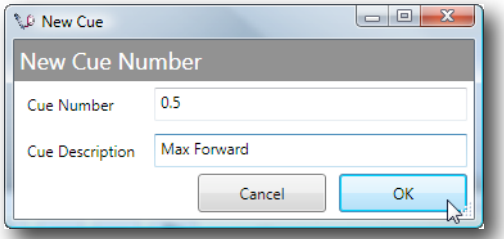
Write two cues

Let's write two cues. The first will send the motor to the maximum forward position, and the second will send it back to the minimum reverse position.

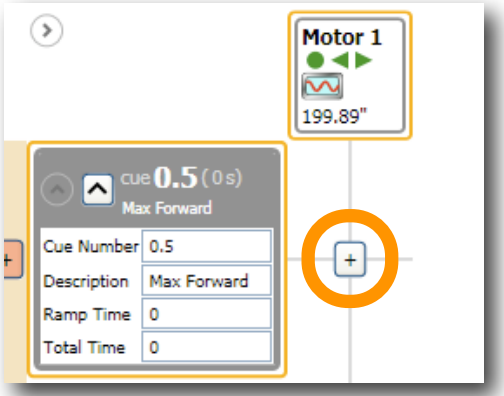
1. There's already a Cue 1 defined by default, so let's add a Cue .5 by right-clicking in the **Cue Grid** and select **Add Cue**. (You could also press **Ctrl + U**, or select **Add Cue** from the **Cues** menu in the menu bar at the top of the screen).



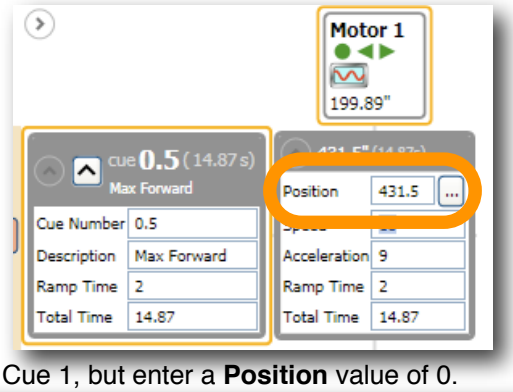
2. A dialog box appears asking for a cue number and a description. Cue numbers can be any positive decimal value (1, 2.34, 1008.38972, etc.), description are just labels you can use to help identify what the cue does. Enter .5 for the **Cue Number** and enter "Max Forward" in the **Cue Description** then press OK.



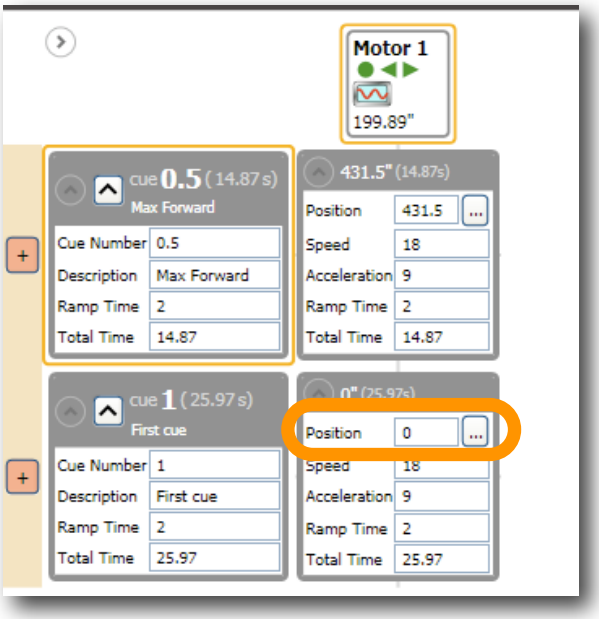
3. Add a movement for Motor 1 in Cue .5 by pressing the "+" button in the **Cue Grid**.



4. Change the **Position** number to match the value you entered as the **Max Forward Position** (in this example it would be 431.5). Leave the speed and acceleration values at their default values.



5. Repeat the process for Cue 1, but enter a **Position** value of 0.

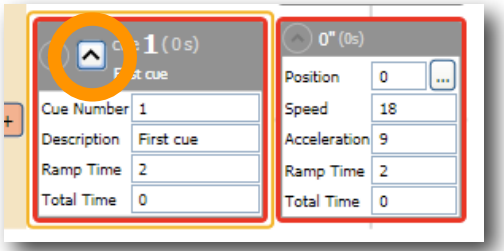


Run the cues

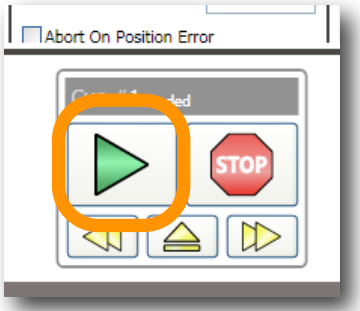
With the two cues written, let's try running the motor.

Depending on your motor system, the cues may or may not work well at this point. We're going to run the cues with the default motor tuning parameters and hope for the best. If the default tuning isn't good enough, the motor won't complete the cue, it will just sit there struggling at very low speed to complete the cue. Don't worry if your cue doesn't run well yet, we'll fix it with motor tuning in the next section.

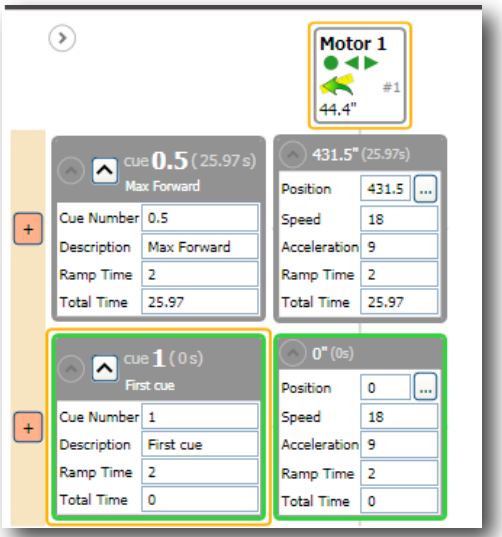
1. To load Cue 1, press the **Load Cue** button on the cue in the **Cue Grid**. The border around the cue box will turn red, indicating that the cue is loaded.



2. To run Cue 1, press the **Run Cue** button in the **Cue Controller** on the right lower side of the screen.



3. As the cue is running, the border around the cue box in the **Cue Grid** turns green.



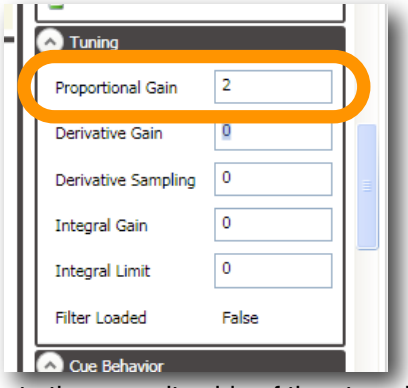
4. When the cue completes, the border around the cue in the **Cue Grid** turns blue. If the border stays green, but the motor is no longer moving this means that the motor tuning needs to be improved. Abort the cue by pressing the **E-Stop** button on your **Showstopper** and go to the next section: *Basic Tuning.*

5. Now run Cue .5 by repeating steps 1 - 4.

Basic tuning

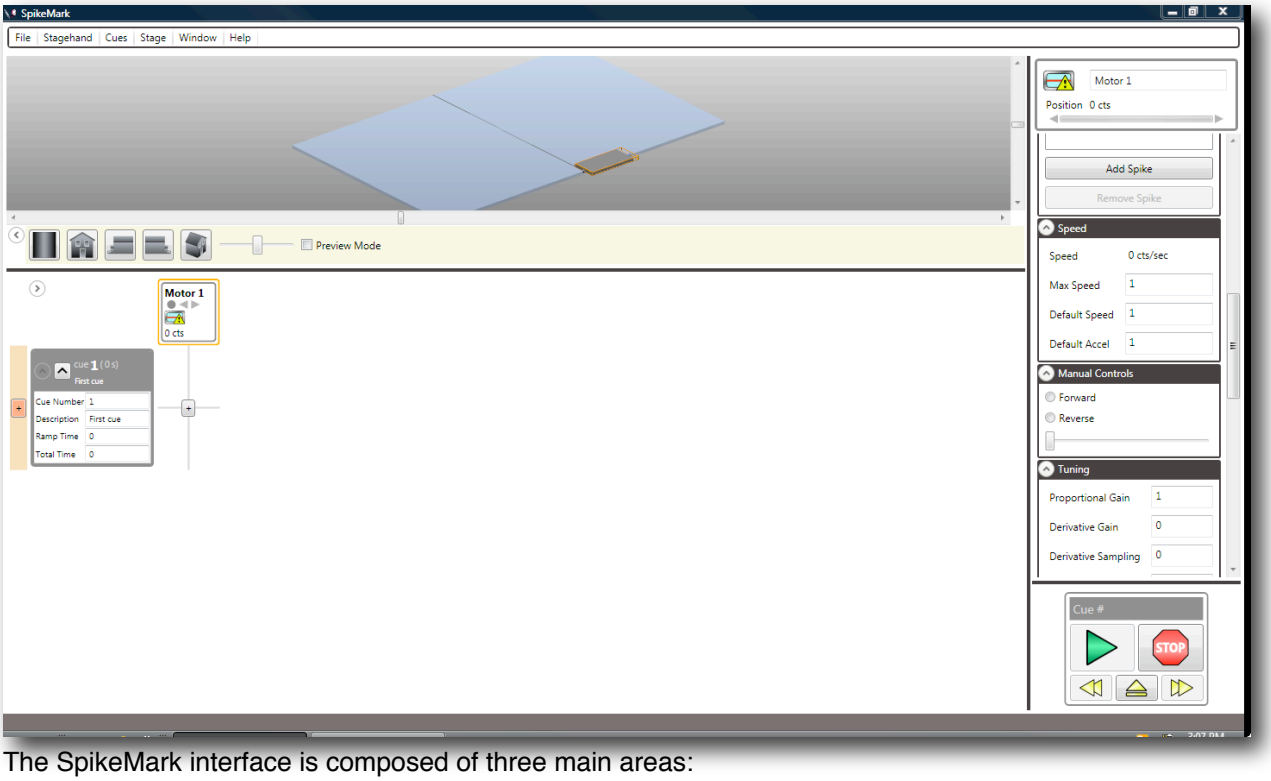
If your first attempt at running cues didn't work and the motor never fully completed the cues, then we need to adjust the tuning parameters. Tuning can be a complex issue, but we're going to take a practical approach in this section without getting bogged down in the details. For more information, see the chapter about **Motor Tuning** later the manual.

1. In the motor editing pane on the right side of the screen, scroll until you find the **Tuning** section.
2. Bump up the **Proportional Gain** value by 1.



3. Run a cue to move the motor to the opposite side of the stage by following the instructions in the previous section.
4. If the cue now completes successfully, you're done. Otherwise repeat steps 1-3 until the motor repeatably completes the cues successfully.

Interface Overview



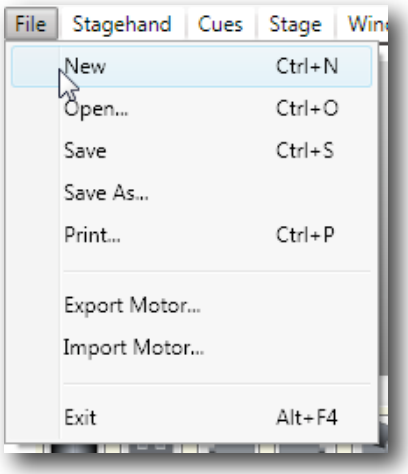
The SpikeMark interface is composed of three main areas:

1. The lower center area contains the **Cue Grid** where cues can be added, removed, and edited.
2. The upper center area contains the **Stage Model** and displays a 3d schematic representation of your theatre and motors.
3. On the right side of the screen there is the **Stagehand Editor** and **Cue Controller**. The **Stagehand Editor** allows setup parameters to be adjusted for both Stagehand Motors & Stagehand FX input/output devices. Beneath the **Stagehand Editor**, the **Cue Controller** mimics the functionality of a physical **Showstopper** giving you the controls needed to run, skip, and stop cues.

The sizes of all these areas can be adjusted by clicking and dragging on the splitter bars that separate each pane. To full collapse a pane, double-click on the splitter bar. To expand a collapsed pane, double-click again on the splitter bar.

Create a New Show

When SpikeMark first launches, it adds a single motor and a single cue to the show. You can start editing the motor and cue as a starting point for your show. If you would like to begin with a blank slate, you can create a new show by selecting **File -> New**.

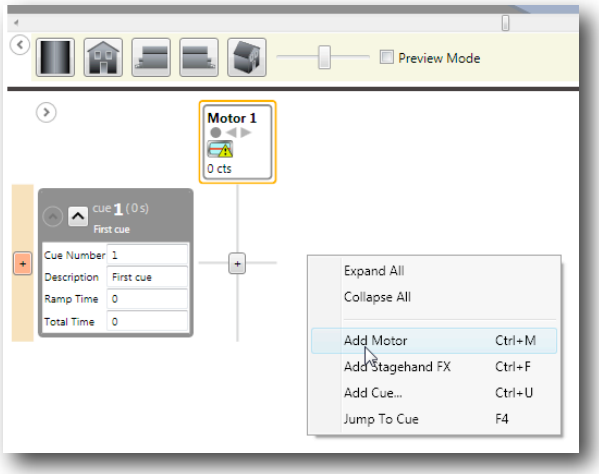


Adding Stagehands

Adding motors

To add a new motor to the show:

1. Right-click in the **Cue Grid** and select **Add Motor.**

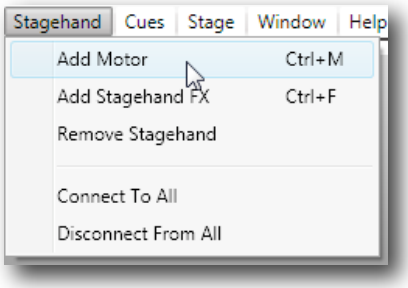


-or-

2. Press **Ctrl** + **M.**

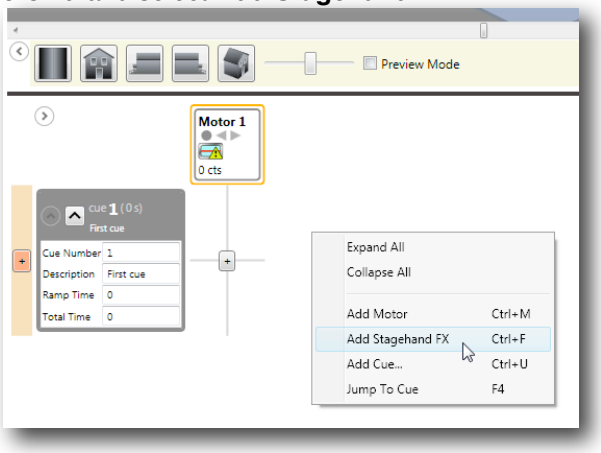
-or-

3. From the menu bar, select **Stagehands** -> **Add Motor.**



Adding Stagehand FX's

1. Right-click in the **Cue Grid** and select **Add Stagehand FX**.

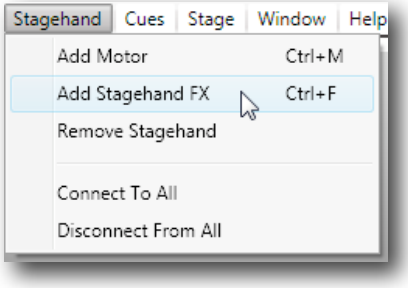


-or-

2. Press **Ctrl** + **F**.

-or-

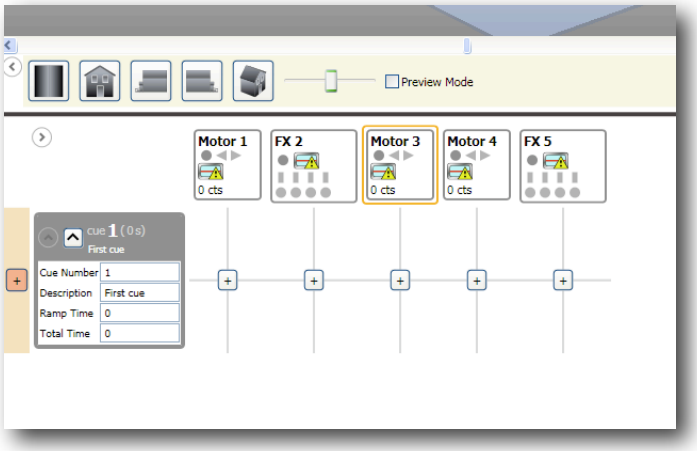
3. From the menu bar, select **Stagehand** -> **Add Stagehand FX**.



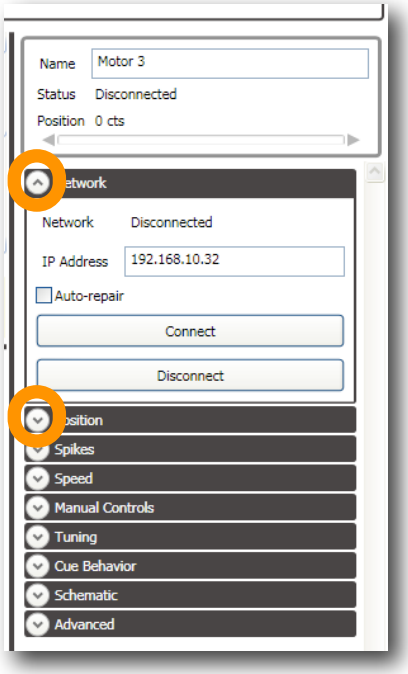
Motor Configuration

Each Stagehand in a show has a collection of parameters that are typically set once during a production. These parameters consist of properties like the name of the motor or FX, it's IP address, it's maximum speed, etc. The **Stagehand Editor** is a pane on the right side of the screen that lets you edit all of the configuration parameters for any motor in the show.

To edit a Stagehand's configuration parameters, select it in the **Cue Grid** by clicking on it. The selected Stagehand will be indicated with an orange outline as shown below.

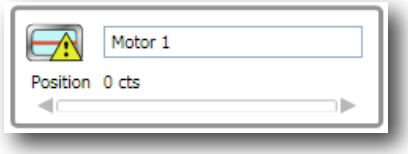


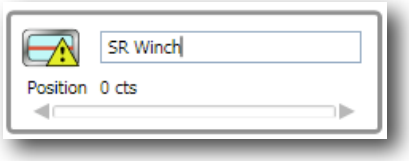
Once you have selected a motor in the **Cue Grid**, you can begin editing it's parameters in the Stagehand Editor. The **Stagehand Editor** is a scrolling window that contains collapsable boxes of parameters grouped by function. You can collapse or expand these boxes to show or hide the information.



Renaming a Motor

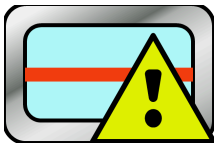
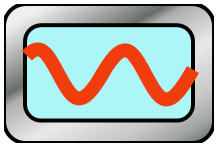
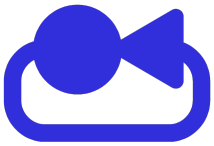
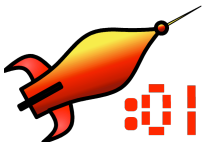
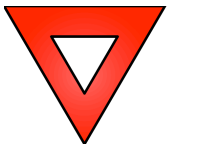
To help you identify all the motors in your show, you can give a motor a name. Typically names reflect the purpose of the motor and make writing cues more intuitive. To change the **Name** of a motor, select the motor to be edited in either the **Cue Grid** or the **Stage Model** and then click into the **Name** field and type a new name. Once you press **Tab** or click out of the field, the new name will be updated in the **Cue Grid**.

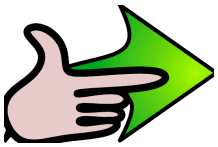
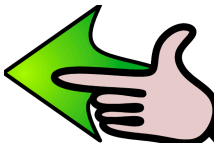
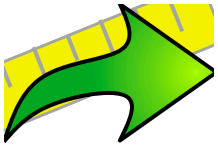
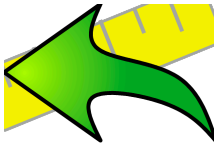
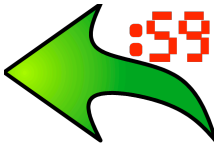




Motor Status

Motor status information is displayed next to the Stagehand name in the Stagehand Editor and in the **Cue Grid** column headers. The status of the motor is indicated by various icons. The different icons are:

Icon	Status
	Motor disconnected from the network.
	Motor connected to the network, but idle.
	Motor has completed a cue.
	Emergency stop.
	Cue loaded.
	Soft stopping, but not yet fully stopped.
	Stopped mid-cue by a soft-stop command.

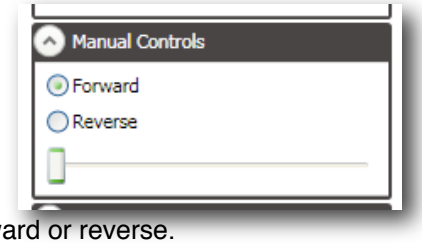
Icon	Status
	In manual mode with the motor enabled, but zero-speed.
	Moving forward in manual mode.
	Moving reverse in manual mode.
	Running a cue forward in target-tolerance mode.
	Running a cue forward in strict-timing mode.
	Running a cue reverse in target-tolerance mode.
	Running a cue reverse in strict-timing mode.
	Stopped by a forward limit switch.
	Stopped by a reverse limit switch.

Icon	Status
	Stopped by a position error fault.

Jogging a Motor Manually

Frequently you will need to move a motor forward or back without running cue. During initial setup, tech, and work calls the scenery often needs to be nudged around the stage. You can either run the motor using the physical jog buttons on the Stagehand, or you can use **SpikeMark**'s on-screen **Manual Controls** to move the motor. To use the on-screen controls:

1. Select the motor you want to move in the **Cue Grid** or **Stage Model**.
2. Confirm that the motor is connected to the network. If it's not connected, establish a network connection.
3. In the **Stagehand Editor** on the right side of the screen, scroll through the options until you find **Manual Controls**.



4. Pick a direction, either forward or reverse.
5. Click and drag the slider slowly to the right to release the brake and ramp up the motor's speed.
6. To stop the motor, slowly move the slider to the left and then release the mouse button. When you release the mouse button the motor will stop and engage the brake.

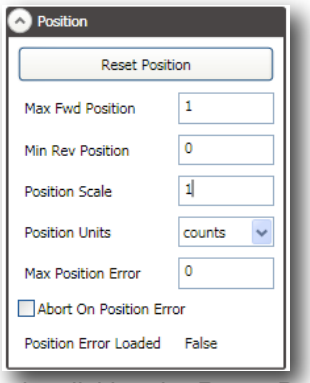
Scaling Motor Position

Whenever a piece of scenery moves, an encoder connected to that scenery spins. As the encoder spins, it generates a stream of electrical pulses. The Stagehand connected to the encoder, counts those pulses to determine where the scenery actually is. The Stagehand relays the position back to **SpikeMark**. **SpikeMark** then displays the position information on screen. By default, **SpikeMark** displays the information directly so that you will view the motor position as raw encoder counts. Viewing the position of the scenery as raw encoder counts is almost never ideal. Usually you'll want to see the motor's position displayed in something more meaningful like inches or degrees.

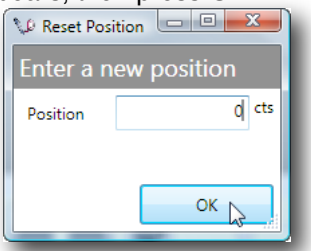
Motor scaling allows you to specify a ratio between raw encoder counts and physical units. With a position scale defined you can view the motor's current position and write cues in a more user-friendly manner. For instance, rather than having to type 1,245,120,295 encoder counts, you will simply enter 123.5".

To determine the correct **Position Scale** for your motor:

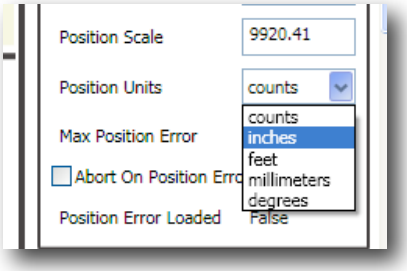
1. Confirm that the current **Position Scale** is set to 1 in the **Stagehand Editor**.



2. Reset the motor's position to zero by clicking the **Reset Position** button.
3. Enter 0 in the dialog box that appears, then press **OK**.



4. Using the **Manual Controls** in the Stagehand Editor, run the motor forward or reverse as far as you safely can.
5. Note the number of encoder counts that were traveled.
6. Measure the distance physically traveled by the scenery.
7. To compute the scaling factor, divide the number of encoder counts traveled by the physical distance *(eg. 1,245,012 / 125.5" = 9,920.41 counts per inch)*.
8. Enter the result from Step 7 into the **Position Scale** field.
9. Select the appropriate units from the **Position Units** pop-up menu.



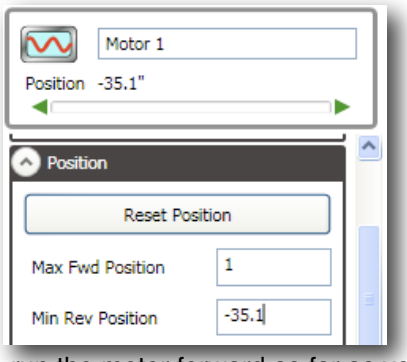
Setting Software Limits For A Motor

All motors in SpikeMark must have a **Max Forward Position** and a **Min Reverse Position** defined in the **Stagehand Editor**. These values are used for two purposes:

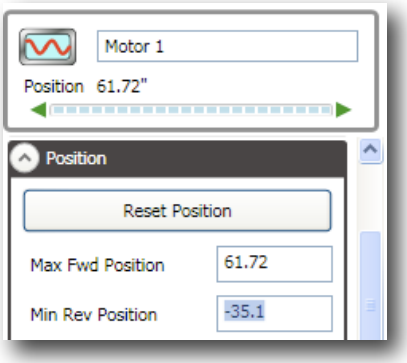
1. **SpikeMark** calculates the track length for all winches in the Stage Model based on these configuration parameters.
2. By setting these values, **SpikeMark** will prevent you from writing cues that travel beyond the physical limits of the machine. If you enter in a position of 1000" for a cue, but the **Max Forward Position** is set to 100", **SpikeMark** will reduce the cue target position to 100" to keep it within the defined limits.

To set your **Max Forward Position** and **Min Reverse Position**:

1. Using the **Manual Controls** in the **Stagehand Editor**, run the motor in reverse as far as you can.
2. Type the **Position** value into **Min Reverse Position**.



3. Using the **Manual Controls**, run the motor forward as far as you can.
4. Type the **Position** value into **Max Fwd Position**.



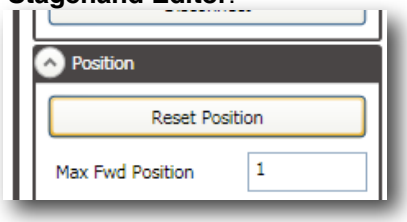
There are a few of rules that you need to follow for **Max Forward Position** and **Min Reverse Position**:

1. The **Max Forward Position** must be greater than **Min Reverse Position**.
2. Values can be negative, provided that rule #1 is followed.
3. Turntables must have software limits, even if there is no physical limitation to how far they can travel. For turntables, often you can just set a high value in both directions (eg. 7200 deg for **Max** & -7200 deg for **Min**)
4. If possible, set your software limits to be slightly less than your physical limit switches.

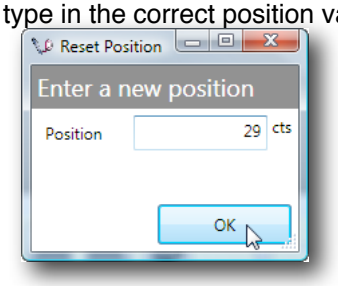
Reset Motor Position

Motor position is stored in the Stagehand, Spikemark simply displays the position information that it receives from the Stagehand. There are times when you need to change the position information that is stored in the Stagehand. For instance, if the Stagehand loses power or is replaced then the position information will be wrong. In such a case, you should move the motor to a known position with the **Manual Controls** on the stage and then reset the position value stored inside the Stagehand. To reset the motor's position value:

1. Select the Stagehand in the **Cue Grid** and confirm that it is connected to the network.
2. Press **Reset Position** in the **Stagehand Editor**.



3. In the dialog box that appears, type in the correct position value and press **OK**.



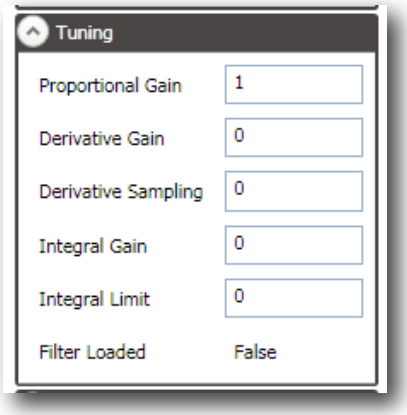
Motor Tuning

Motor tuning is perhaps the most difficult aspect of scenic automation. By tuning a motor, you are describing to a Stagehand how it should regulate the motor speed during a cue using mathematical formulae. We won't get too deep into the intricacies of that math, but it is hard to discuss motor tuning without getting a little involved in the nuts and bolts of motion control theory.

First, it helps to understand the problem that motor tuning solves. My favorite analogy for motor tuning is cruise control on a car. When you set your cruise control to 55 mph (as I'm sure you do, being a law-abiding citizen), the car has to adjust the amount of fuel going to the engine to keep moving at that speed. If you are going up hill, the car lags for a second and then more gas rushes into the engine and the car speeds up to get back to the desired speed. Motion control for scenery is very similar. We program the Stagehands with a speed and target position, and the Stagehand has to adjust power to the motor to achieve the desired effect. If there's a bump in the floor, the winch has to have a little more power to maintain speed. Of course, if the winch has too much power, it will lurch over the bump and then quickly reduce power so that it doesn't go too fast once it is over the bump.

The Stagehand is constantly analyzing where the motor is versus where it should be and then adjusting motor power to minimize the difference between where the motor should be and where it really is. It does this analysis a few million times per second. When it wants to apply power to correct for error in position, it looks to us for guidance. By entering in some tuning parameters, we are giving the Stagehand that guidance. In a confounding abstract way, we are specifying how much power to give the motor when it needs to make a correction. If the values that we enter give the motor too much power during correction, the motor will be jerky as it over-corrects and then has to pull back (remember, this happens millions of times per second). If the values we enter do not provide enough power to the motor to correct position, it will never reach the cue position since it will run out of power and be unable to muscle the load onto the target.

There are five tuning values that can be used to achieve smooth, accurate motion. To adjust these parameters, select the motor you wish to tune in the **Cue Grid** and then scroll to the **Tuning** section of the **Stagehand Editor**.



- **Proportional Gain** is a ratio of power to position error. The higher this value, the more power will be applied to correct for position errors. This is often the only tuning parameter you will need to adjust. Proportional Gain needs to be at least 1 for the motor to move at all. If the value is too high, the motor will begin lurching. Sometimes this lurching can be extreme so make sure to have your hand on the E-Stop when you first adjust this parameter.

- **Derivative Gain** is applied against changes in position error. It can have a dampening effect on a motor, absorbing the shocks from a high Proportional Gain.

- **Derivative Sampling** describes how often the Derivative Gain should be applied. If you are using Derivative Gain, Derivative Sampling must be at least 1. I rarely see any benefit to using a Derivative Sampling value higher than 2.

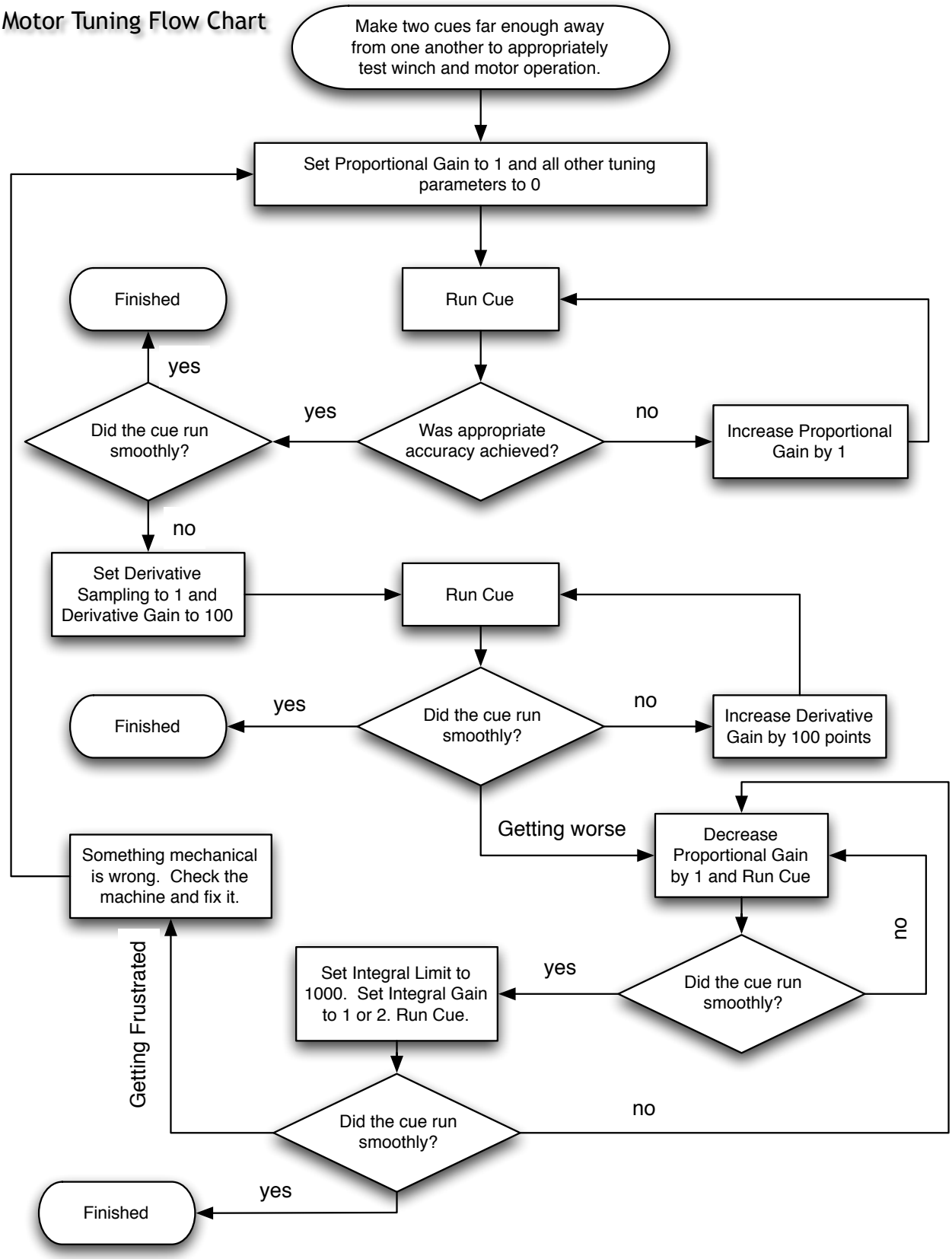
- **Integral Gain** tries to compute corrective action by evaluating the position error and motor response over time. From a practical standpoint, Integral Gain will often correct the motor position at the end of travel. This can be helpful if you need to reduce **Proportional Gain** to smooth out a motion, but then loose accuracy. In such a scenario, adding in 1 or 2 points of Integral Gain will often retain accuracy without sacrificing smooth travel.

- **Integral Limit** prevents a condition known as Integral Windup, which can lead to really erratic motor behavior. If you use an **Integral Gain**, set Integral Limit to 1000.

- **Filter Loaded** is not an adjustable tuning parameter, rather it indicates whether the filter parameters have been sent to the motor yet. After you change a tuning parameter, that change is saved until the next cue is run. When the next cue is run, the new filter parameters will be sent to the motor automatically.

Now you know the basic theory of motor tuning and a definition for each tuning parameter. But how do you actually tune a motor?! Over the past several years, we've come up with a practical approach to motor tuning that yields good results in a reasonable amount of time. That approach is shown on the following pages in the form of a flow chart. The steps are clear and simple, but it can still be a frustrating experience the first time you attempt to tune a motor. I can't stress enough how important it is to find at least an hour of quiet time on the stage without anyone else working around the moving scenery. Get a cup of coffee, relax, focus and take a stab at it. If that doesn't work, give us a call. We have had plenty of experience helping people tune motors over the phone. The good news is that it gets a lot easier once you see it work the first time.

Motor Tuning Flow Chart


Make two cues far enough away from one another to appropriately test winch and motor operation.
Set Proportional Gain to 1 and all other tuning parameters to 0
Run Cue
Was appropriate accuracy achieved?
no
Increase Proportional Gain by 1
yes
Did the cue run smoothly?
yes
Finished
no
Set Derivative Sampling to 1 and Derivative Gain to 100
Run Cue
Did the cue run smoothly?
yes
Finished
no
Increase Derivative Gain by 100 points
Getting worse
Decrease Proportional Gain by 1 and Run Cue
Did the cue run smoothly?
yes
Set Integral Limit to 1000. Set Integral Gain to 1 or 2. Run Cue.
no
Did the cue run smoothly?
yes
Finished
no
Getting Frustrated
Something mechanical is wrong. Check the machine and fix it.

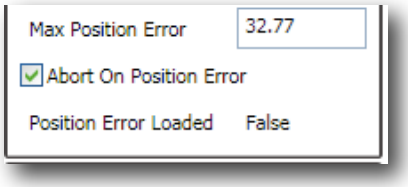
Max Position Error

Once a motor has been adequately tuned, it is a good idea to turn on **Abort On Position Error** and set a **Max Position Error** value. This protects you from runaway motors when the encoder is damaged or disconnected. Without a **Max Position Error**, the Stagehand will send an increasing amount of power to compensate for a position error. If the encoder is not sending information to the Stagehand, the Stagehand will think that the motor is not moving. So it will rapidly work up to full speed and run at full speed indefinitely, since, from the Stagehand's perspective, the motor is standing still. Without setting **Max Position Error**, you will be forced to use the Emergency Stop but not before you have been scared out of your skin.

On the other hand, a **Max Position Error** value can stop a motor when it is poorly tuned, but not posing any danger. This interrupts the show and keep be detrimental to the performance. To keep the motor from faulting during normal circumstances, the default value for maximum position error is set as high as possible. You can tighten the allowable position error by lowering this value if you have requirements for a more exacting motion profile.

To set the maximum position error:

1. Select the motor you want to adjust in the **Cue Grid**.
2. Check the **Abort On Position Error** option in the **Position** section of the **Stagehand Editor**. If you like, you can tweak the value of the Max Position Error (this value is given in scaled position units).



Note: Position error changes are loaded into the Stagehand prior to running a cue.

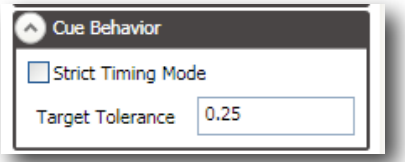
Motor Cue Behavior

SpikeMark has two modes for running motor cues:

1. **Strict Timing** mode insists that the motor move accurately and achieve the cue position within the theoretical time allowed for that movement based on the programmed distance, speed, and acceleration. If the motor cannot achieve target in the computed time, it will be turned off regardless of its current position. For shows that have tight timing requirements, this option may be useful. However, it has mostly been included in SpikeMark since it was the only mode available in earlier versions of Avista. When using this mode, you have to be diligent about tuning your motors well. Sloppy tuning will result in motors abruptly stopping in mid-move.
2. **Target Tolerance** mode is new in SpikeMark and has been implemented to make motor cues more reliable regardless of poor tuning. When using **Target Tolerance** mode, the motor will continue to try to achieve target regardless of how much time has passed. Once the motor position is within your specified tolerance, it will turn off and display a **Complete** status icon. **Target Tolerance** mode is the default mode in **SpikeMark**.

To specify which mode to use:

1. Select the motor you wish to edit in the **Cue Grid** and scroll to the **Cue Behavior** section in the **Stagehand Editor**.



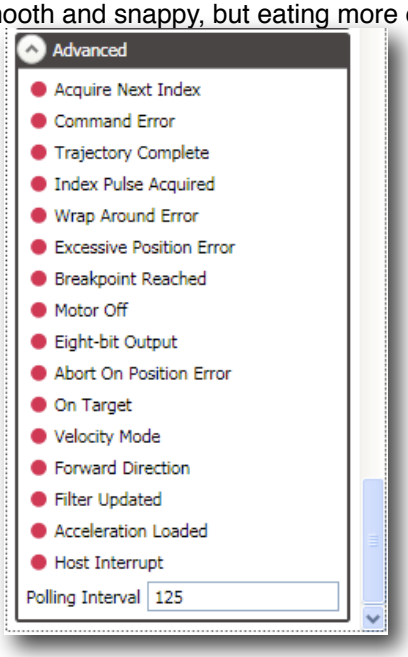
2. To use **Target Tolerance** mode, make sure that **Strict Timing Mode** is unchecked and enter a **Target Tolerance**. Tolerance values are in scaled position units (inches, degrees, etc.).
3. To use **Strict Timing Mode,** check the box. When **Strict Timing Mode** is checked, the **Target Tolerance** value will be ignored.

Motor Refresh Speed

Whenever **SpikeMark** connects to a motor, it starts asking the Stagehand for updated information about it's status, position, limits, and e-stop condition. SpikeMark asks for updated information several times a second to give the illusion of a steady stream of information. This is how the position information appears to fluidly change as the motor moves on stage. Most of the time, this all works great and you don't have to think about *how* it works. However, if you have a large show, or a slow computer, you may find that **SpikeMark** seems to be slowing down or feeling clunky. In such a case you may wish to tweak the frequency of information updates, known as the **Polling Interval**, so that SpikeMark doesn't get bogged down with this constant updating process.

To change the polling interval:

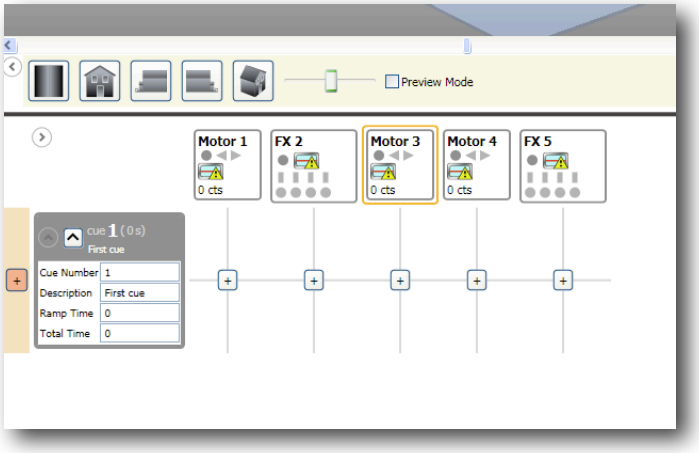
1. Select the motor you want to adjust in the **Cue Grid** or **Stage Model**.
2. Scroll through the **Stagehand Editor** until you find the **Advanced** section and expand it.
3. At the bottom of the **Advanced** section you can enter a new **Polling Interval**. The values are in milliseconds. The default is 125, which means that SpikeMark updates motor information 8 times a second. Setting the **Polling Interval** to 250 will update motor information 4 times a second, which will relieve the computer of some processing stress but the updates on screen will appear more jerky. Setting the polling interval to 50 will provide updates 20 per second making the information updates very smooth and snappy, but eating more of your computer's resources.



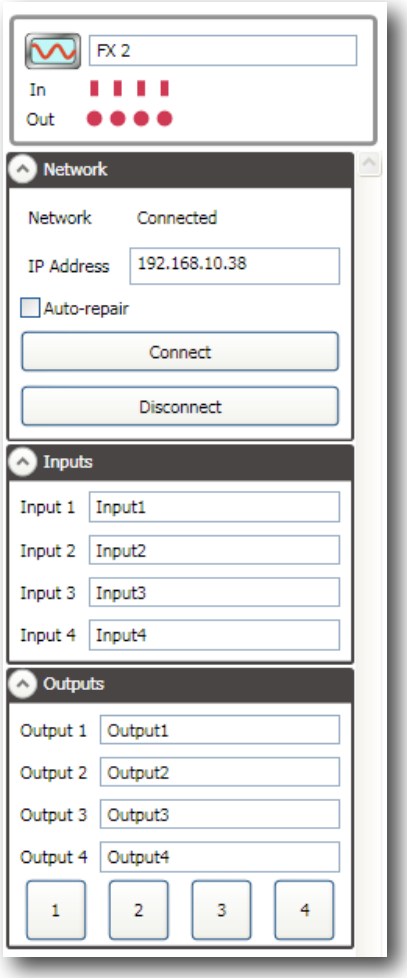
Stagehand FX Configuration

Each Stagehand in a show has a collection of parameters that are typically set once during a production. These parameters consist of properties like the name of the motor or FX, it's IP address, it's maximum speed, etc. The **Stagehand Editor** is a pane on the right side of the screen that lets you edit all of the configuration parameters for any motor in the show.

To edit a Stagehand's configuration parameters, just select it in the **Cue Grid** by clicking on it. The selected Stagehand will be indicated with an orange outline as shown below.



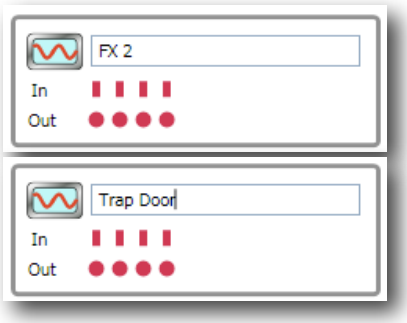
Once you have selected a motor in the **Cue Grid**, you can begin editing it's parameters in the **Stagehand Editor.** The **Stagehand Editor** is a scrolling pane that contains collapsable boxes of parameters grouped by function. You can collapse or expand these boxes to show or hide the information.



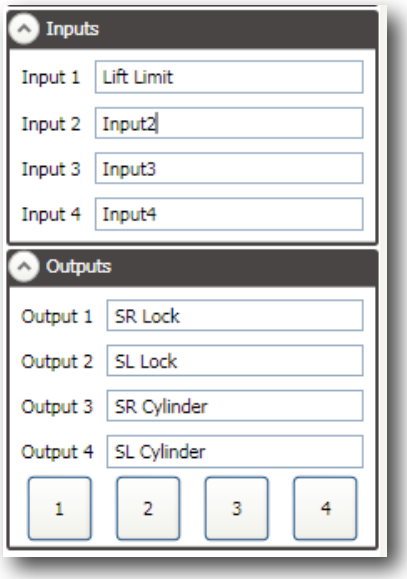
Renaming an FX

To help identify all the Stagehand devices in your show, you can give an FX any name you want. Additionally, you can rename all of the Inputs & all of the Outputs for an FX to help remember the purpose of each switch in the Stagehand FX.

To change the **Name** of an FX, select the Stagehand to be edited in either the **Cue Grid** and then click into the **Name** field and type a new name. Once you press **Tab** or click out of the field, the new name will be updated in the **Cue Grid**.

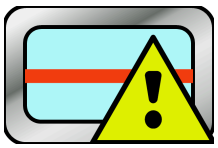
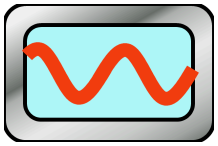
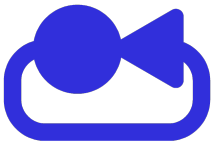


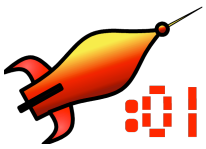
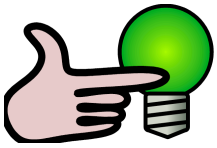
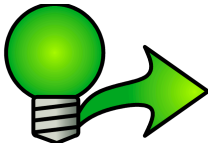
To rename the inputs or outputs for an FX, type in a new name in any of the Input or Output fields.



FX Status

Stagehand FX status information is displayed next to the Stagehand name in the Stagehand Editor and in the **Cue Grid** column headers. The status of the FX device is indicated by various icons. The different icons are:

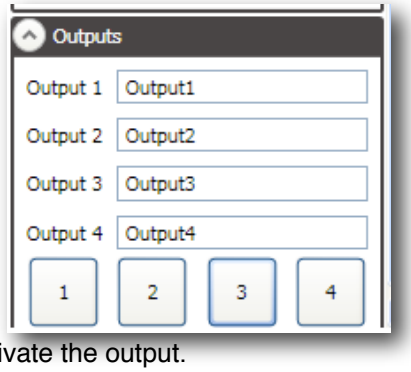
Icon	Status
	FX disconnected from the network.
	FX connected to the network, but idle.
	FX has completed a cue.

Icon	Status
	Emergency stop.
	Cue loaded.
	Soft stopping, but not yet fully stopped.
	Stopped mid-cue by a soft-stop command.
	An output is active in manual mode.
	One or more outputs is running in a cue.
	All outputs have been turned off because of a fault condition.

Manually Activating FX Outputs

If you want to test an FX output switch, you can use the manual output buttons in the **Stagehand Editor**. Press the button whose number corresponds to the output you wish to activate. The button behaves like a momentary switch, the fx output will be active as long as the button is held down. When you release the button, the fx output will deactivate.

1. Select the Stagehand FX you want to activate in the **Cue Grid**.
2. Press one of the output buttons in the **Outputs** section.

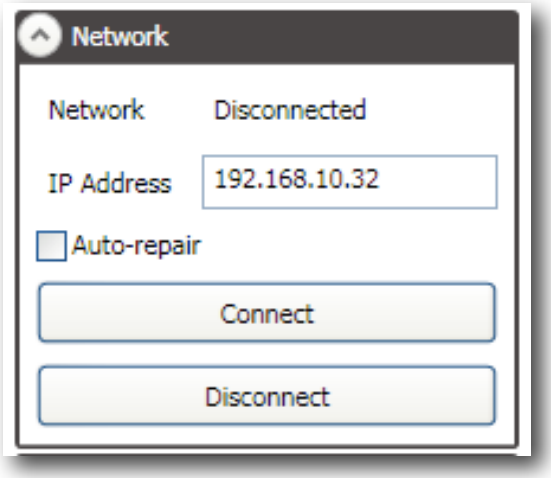


3. Release the button to deactivate the output.

Network Connections

Editing Stagehand Network Settings

The Network group inside the **Stagehand Editor** contains the settings that pertain to Ethernet connection between SpikeMark and the selected Stagehand.



1. **Network** - The first field shows the current status of the network connection. The possible values are:
   - Disconnected - The network connection has not been made yet.
   - Connecting - A connection is trying to be established.
   - Errors - The connection failed. Connections can fail for various reasons, but commonly: a cable was removed, the specified Stagehand address is wrong, or the computer running SpikeMark has an incompatible address (see the **Quick Start** chapter for steps to configure your computer's IP address).
2. **IP Address** - This field contains the IP address of the selected Stagehand that you wish to connect to. Enter the IP address exactly how it is displayed on the Stagehand's face panel. Valid address range from 0.0.0.0 to 255.255.255.255, but remember that the first three segments of the address must match between the computer running SpikeMark and all the Stagehands on the network, while the last segment must be unique for each device.
3. **Auto-repair** - If this option is checked, SpikeMark will automatically try to reconnect to a Stagehand if the connection is lost for any reason. Most of the

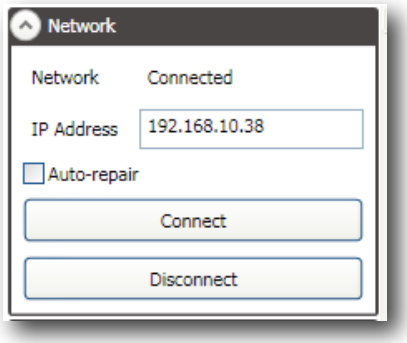
time you'll turn this option on for running shows, but you may want to turn it off during initial set up to take a motor 'offline'.

4. **Connect** - Press this button to try connecting to the Stagehand.
5. **Disconnect** - Press this button to disconnect from the Stagehand.

Connecting to a Stagehand

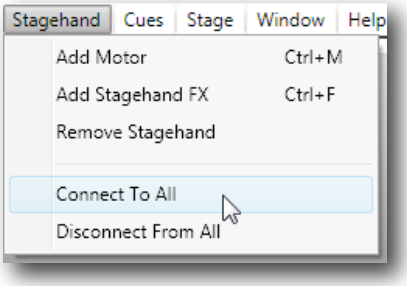
All communications between **SpikeMark** and Stagehands happens over the network. To run any cues, move any motors, or activate an FX outputs, a network connection must be made to the Stagehand you wish to control. The connection only needs to be established once per show, unless the network connection is lost unexpectedly. There are 3 ways to establish a network connection.

1. Select the Stagehand you wish to connect to from the **Cue Grid** and press the **Connect** button inside the **Stagehand Editor**.



-or-

2. You can connect to all the Stagehands in the show at once by selecting **Stagehand -> Connect To All** from the menu bar.



-or-

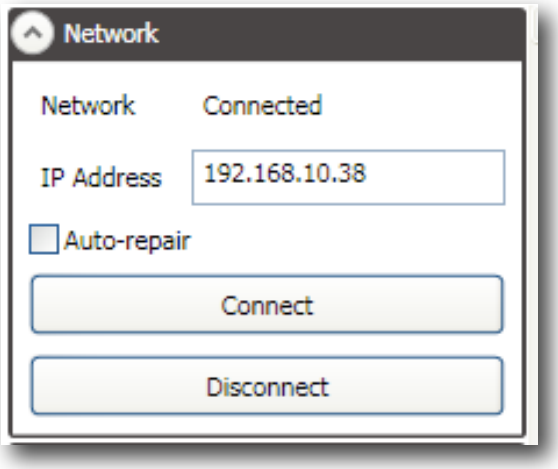
3. Each time you open a show file, SpikeMark will automatically attempt to connect to all the Stagehands in the show.

Disconnecting from a Stagehand

You can break the network connection to a Stagehand any time you want to temporarily remove it from **SpikeMark**'s control. Sometimes during rehearsal you may want to run some cues without moving one of the motors. In such a scenario, breaking the network connection is the fastest way to temporarily remove the motor from the cue. There are two ways to break a network connection.

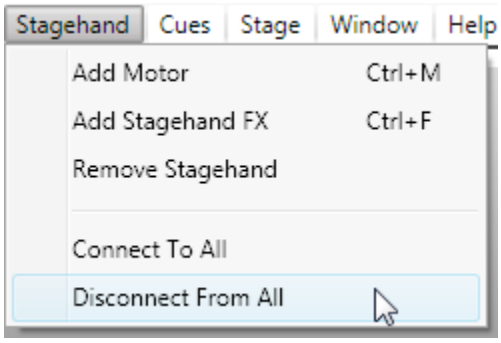
*Note: Make sure you uncheck **Auto-repair** in the **Stagehand Editor** if you want to disconnect from a Stagehand, otherwise SpikeMark will immediately try to re-connect.*

1. Select the Stagehand you want to disconnect in the **Cue Grid** and press the **Disconnect** button in the Stagehand Editor.



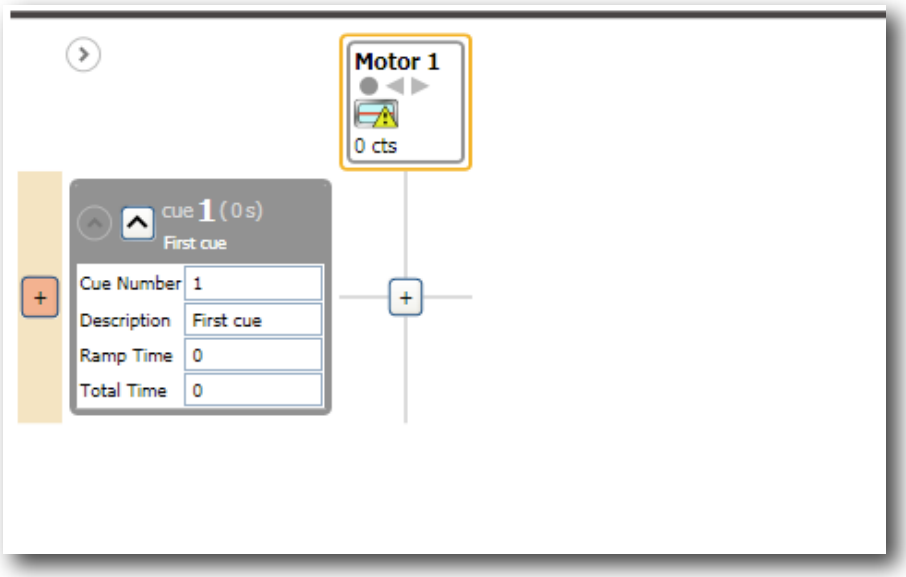
-or-

2. To disconnect from all Stagehands in the show select Stagehands -> Disconnect From All in the menu bar.



Writing Cues

A Cue is a collection of movements including any number of motors or FX outputs that all start when you press the GO button on the Showstopper or on-screen **Cue Controller**. You can have an unlimited number of cues to your show. Each cue has a cue number that can be any decimal number, allowing for 'point cues' (eg. 1.5, 209.52, etc.). SpikeMark displays all the cues in the show in the Cue Grid as rows sorted by cue number. When **SpikeMark** is first launched, a single cue, Cue #1, is created by default. You can use this cue as a starting point or delete it and start with a blank slate.

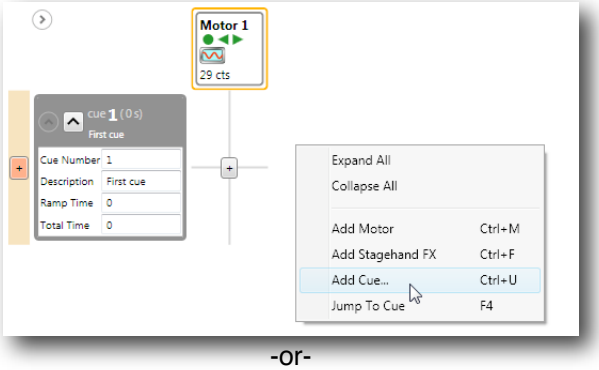


Note: It is worth mentioning that you can write cues with or without being connected to any Stagehands. If desired, you can "rough in" all of your cues before setting up your motors on stage. By using the Spikes feature and a scaled ground plan, you can write all of the cues for the show and then tweak final positions when the scenery is installed.

Adding Cues

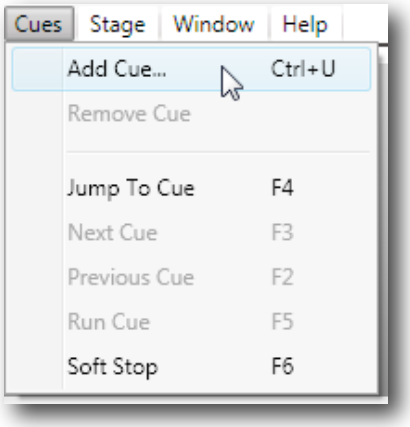
There are three ways to add cues to your show.

1. Right-click on a blank area in the **Cue Grid** and select **Add Cue...**



-or-

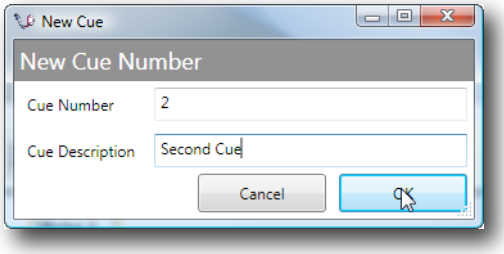
2. Select **Cues -> Add Cue** from the menu bar.



-or-

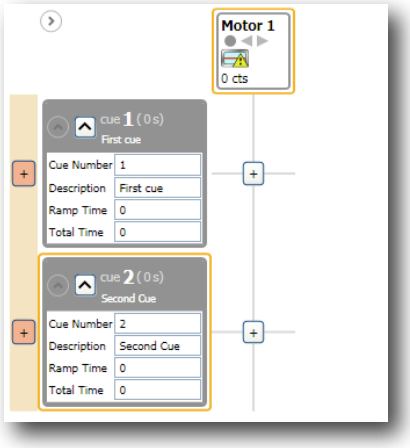
3. Use the keyboard shortcut **Ctrl** + **U**.

A dialog appears prompting you for a cue number and description.



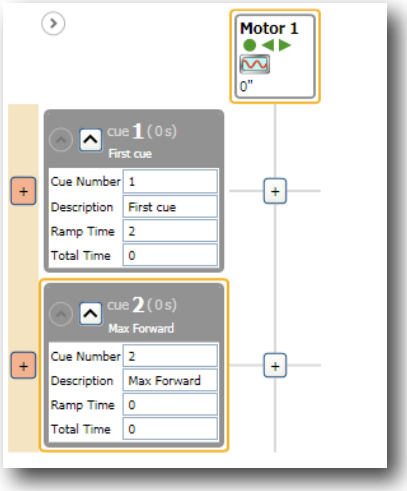
You can edit the **Cue Number** or accept the default. The **Cue Description** is solely for your benefit. You can enter a descriptive name to help identify the cue in the **Cue Grid**. Press **OK** to accept the new cue, or **Cancel** to discard it.

The newly created cue will appear in the **Cue Grid**.

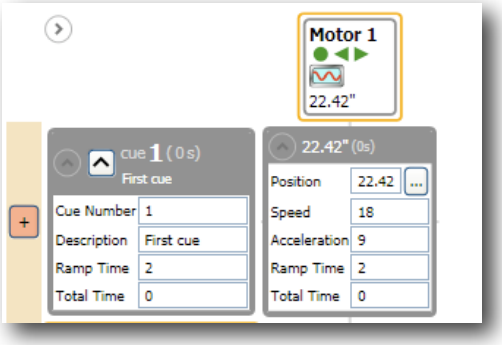


Adding Motor Movements To A Cue

You can add a motor movement to a cue by pressing the “+” button that is displayed at the intersection of the cue row and the motor column.



After you press the "+" button, a movement will appear displaying the parameters for the motor in the cue.



A motor movement has the following parameters:

1. **Position** : The target position where you want the motor to go. This number is displayed in scaled units (inches, feet, degrees, etc.). By default, SpikeMark uses the current position of the motor as the target position. This makes it easy to write cues by jogging the motor into the desired position and then adding the movement to record the 'look' on stage. The button to the right of the **Position** text box will bring up the **Spike Selection** window. Spikes are discussed later in this chapter.
2. **Speed** : The speed at which you want the motor to travel. This number is displayed in units per second (inches/sec, degrees/sec, etc.). You can synchronize multiple motors by programming them at the same speed, regardless of gearing and encoder differences. Valid speed values are greater than zero and less than the **Max Speed** defined in the **Stagehand Editor**. SpikeMark initially uses the **Default Speed** defined in the **Stagehand Editor**.
3. **Acceleration** : The rate at which the speed will increase as the motor moves. This number is displayed in units per second per second (inches/sec/sec, etc.). A higher value will cause the motor to accelerate more quickly giving the motion more snap at the beginning and end of the cue. A lower value will cause the motor to accelerate more slowly and smoothly. Valid acceleration values range from greater than zero to no greater than the programmed **Speed** value. SpikeMark initially uses the **Default Acceleration** defined in the **Stagehand Editor**.
4. Ramp Time : The number of seconds that the motor will take to accelerate up to programmed speed. If you would like to program the cue in time units, you can edit this field rather than using **Acceleration**. SpikeMark will automatically calculate a new **Acceleration** based on the **Ramp Time** you enter.

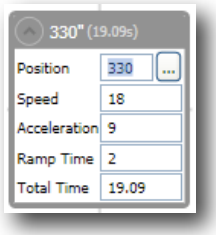
5. **Total Time** : The total number of seconds that the movement will take to complete. If you know that the motor needs to move the scenery in 10 seconds, you can enter a value of 10 here and SpikeMark will recalculate the **Speed** and **Acceleration** values to make that happen.

Note: SpikeMark computes the Total Time and Ramp Time in the first cue of the show by comparing the programmed position to the motor's current position. In all subsequent cues in the show, SpikeMark compares the programmed position to the previous cue's programmed position.

Change Motor Position In A Cue

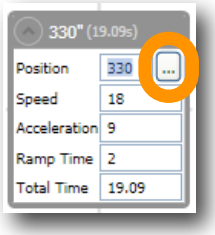
To change the programmed position of a motor in a cue you can do either of the following:

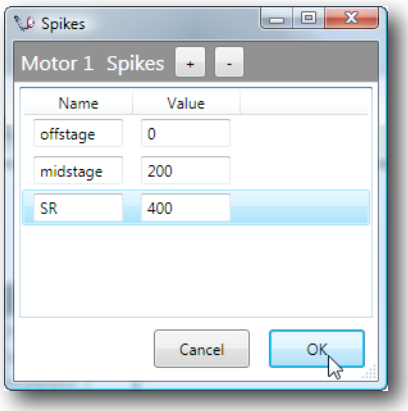
1. Type a new position value directly into the **Position** text box. After you enter a new **Position**, SpikeMark will recalculate the **Total Time** of the movement based on the new target position. If you had previously specified a **Spike** for the movement it will be removed and replaced with the explicit value entered.



-or-

2. Press the **Spike** button next to the **Position** box and select a **Spike** from the list. Press OK to accept.

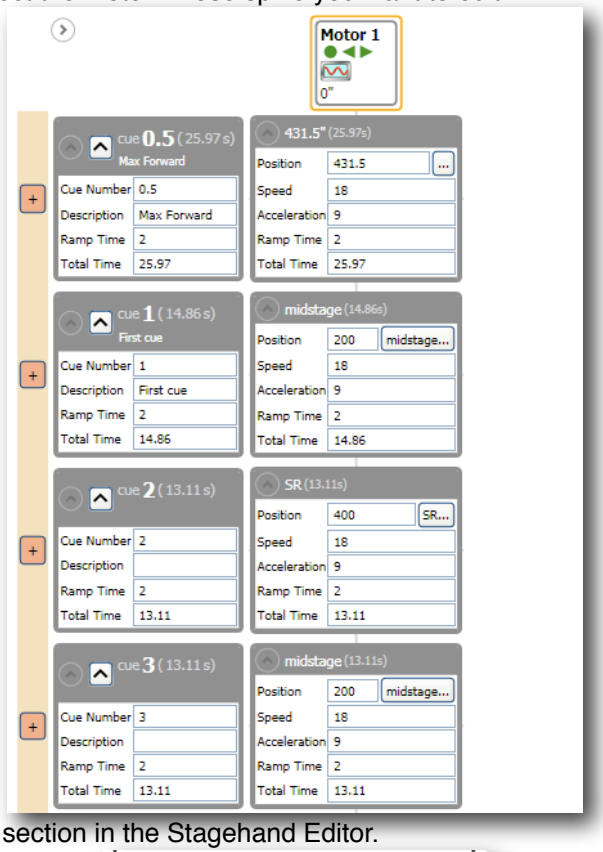




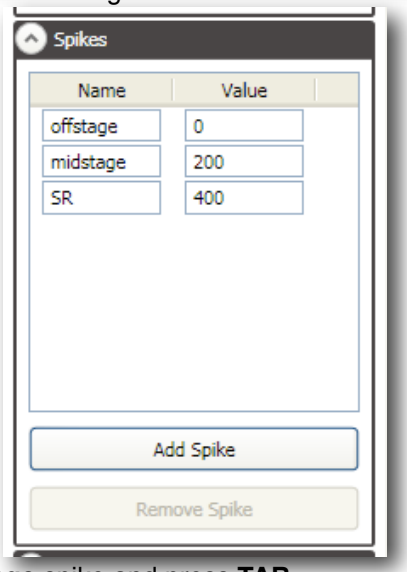
Change A Spike Position

A key advantage to using spikes is that you can edit the spike value and all cues that reference the spike will be updated. This makes it very easy to tweak position values and not have to track the changes manually through the entire show. For example, we've defined a spike named **midstage**. However, during the rehearsal process it becomes evident that the **midstage** position needs to move onstage a quarter of an inch. We've already used this spike in a bunch of cues, so to update the spike and have the new position updated in all cues we just edit the spike.

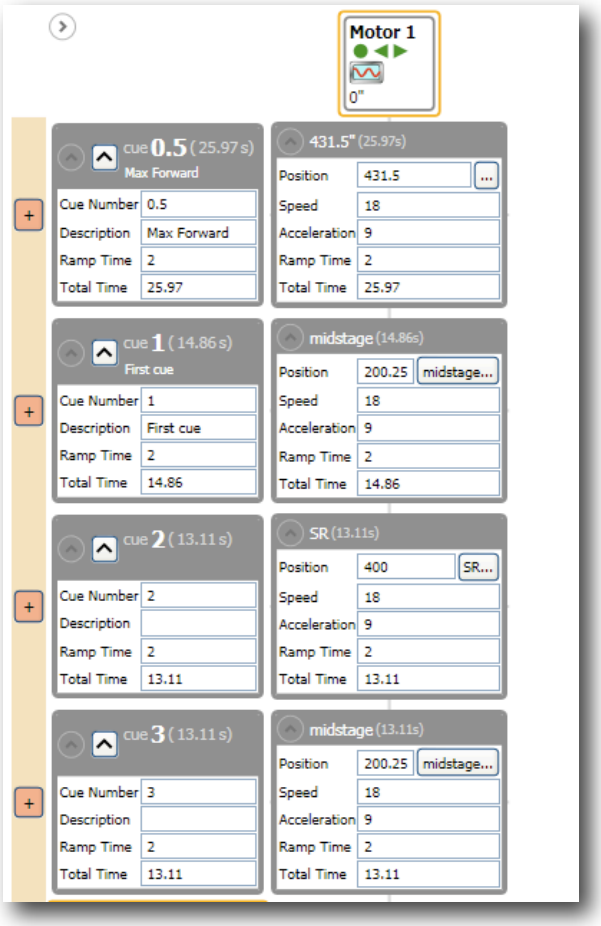
1. In the Cue Grid, select the motor whose spike you want to edit.



2. Scroll to the Spikes section in the Stagehand Editor.



3. Edit the value for the **midstage** spike and press **TAB**.
4. Note that the position values have been updated in cues that reference **midstage** spike.

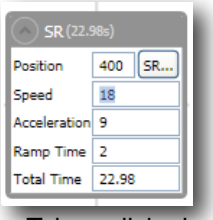


Change Motor Speed & Time In A Cue

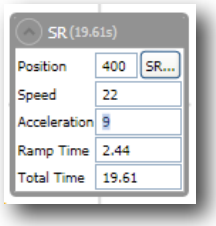
There are two ways to adjust how fast a motor moves. You can either specify the speed that you want the motor to move, or you can specify how long it should take for the motor to reach the target position and let **SpikeMark** compute the speed.

To change the speed of a motor directly:

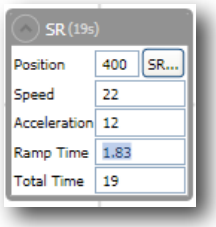
1. Select the old speed value.



2. Enter in a new Speed value and press Tab or click elsewhere to accept the new value. Note that **SpikeMark** computes the new **Total Time** & **Ramp Time** for the cue.

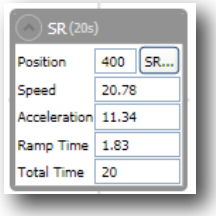


3. Repeat the process for the **Acceleration** value and note that **Total Time** & **Ramp Time** are recomputed.

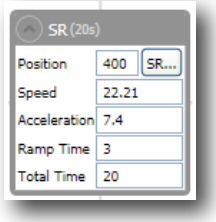


To change the duration of the movement:

1. Select and change the **Total Time** value. Note that the **Speed** and **Acceleration** are recomputed.



2. Select and change the **Ramp Time** value. Note that **Speed** and **Acceleration** are recomputed.

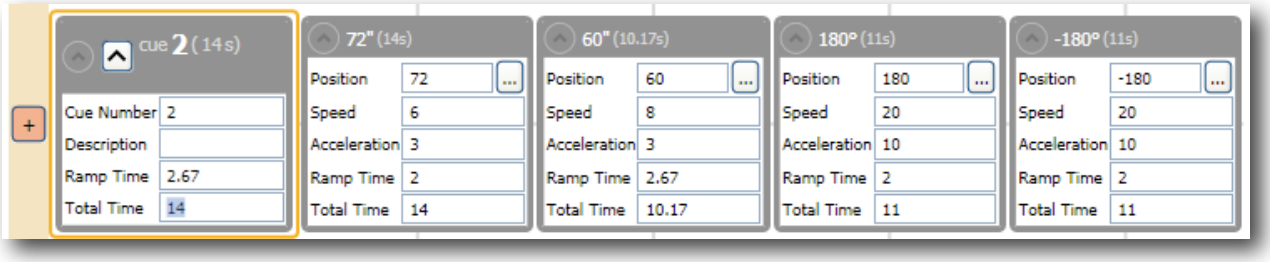


Note: If you enter a Total Time or Ramp Time combination that would require the motor to exceed its defined Max Speed, Total Time and Ramp Time will be recomputed to be as fast as possible.

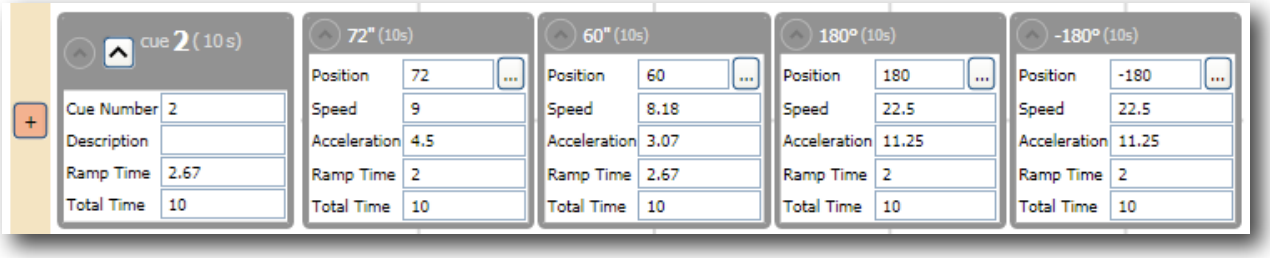
Changing Cue Time

You can change the time for all the movements in a cue by specifying the **Total Time** and **Ramp Time** for the entire cue. This makes it easy to write cues that have all the motors ending at the same time.

1. Select the **Total Time** for the cue you want to edit in the **Cue Grid**.



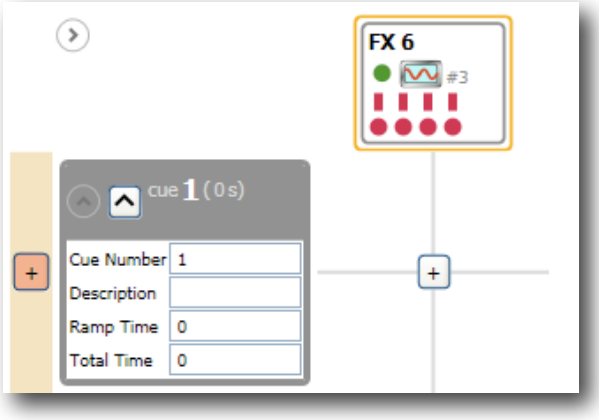
2. Type in the new **Total Time** value you want to use and press **TAB**. Note that all the **Total Time** value for each movement has been updated.



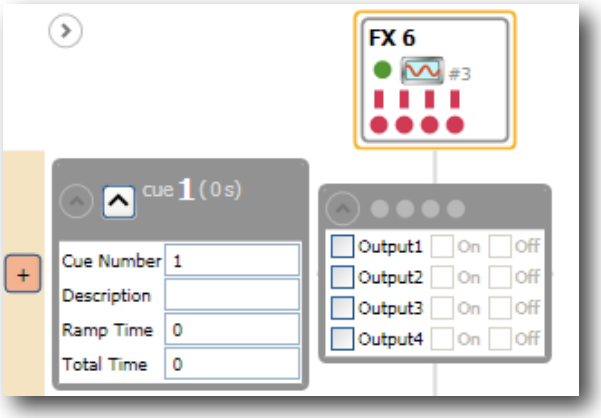
3. Repeat the steps for **Ramp Time**, if desired.

Adding an FX Movement

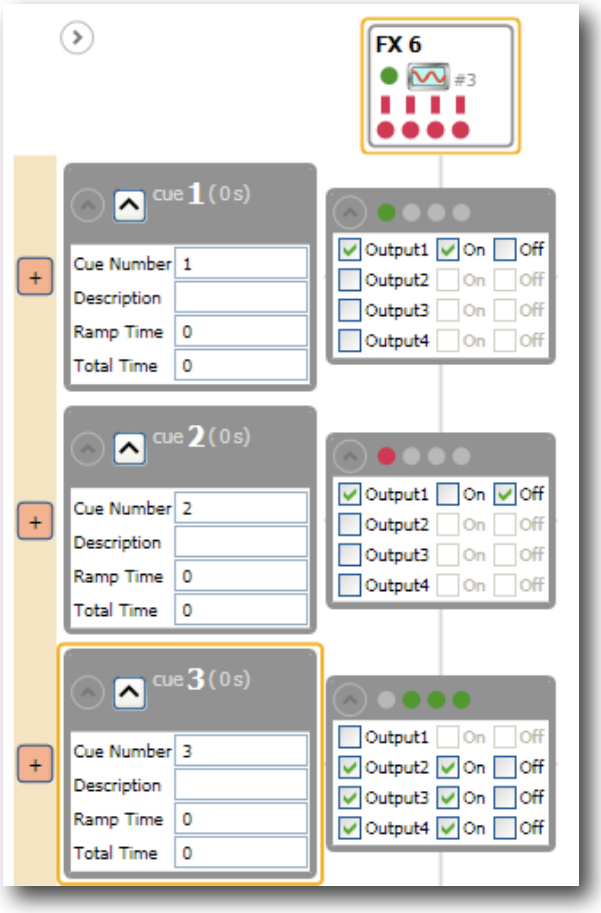
Adding an fx movement is the similar to adding a motor movement. Click the “+” button in the **Cue Grid** that is aligned with the correct Stagehand FX and Cue.



An fx movement will appear in the **Cue Grid**.



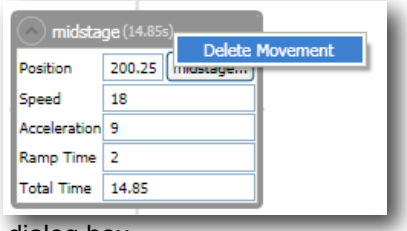
An fx movement can turn any output switch on or off in the cue. If a switch is turned on in one cue, it will remain on until another cue turns it off. If you don't want to change the state of a switch in a cue, leave the switch unchecked completely. Below is an example series of cues that turns Output 1 on, then off, then turns Outputs 2-4 on while leaving Output 1.



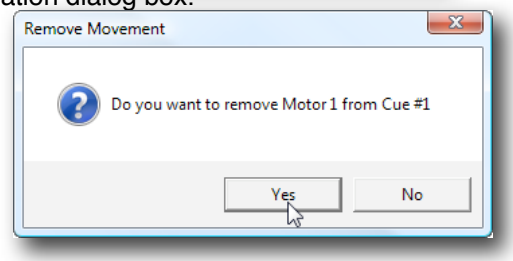
Deleting a movement

Sometimes you will right a cue with a bunch of motors moving and then discover one of the motors shouldn't be in the cue. To remove a movement from a cue:

1. Right-click on the movement you want to delete in the **Cue Grid** and select **Delete Movement**..



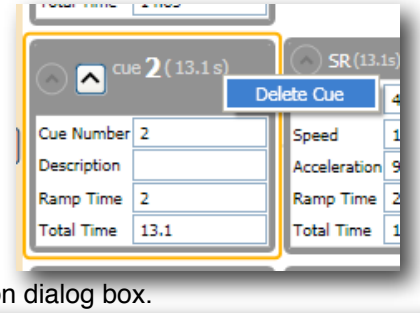
2. Click Yes in the confirmation dialog box.



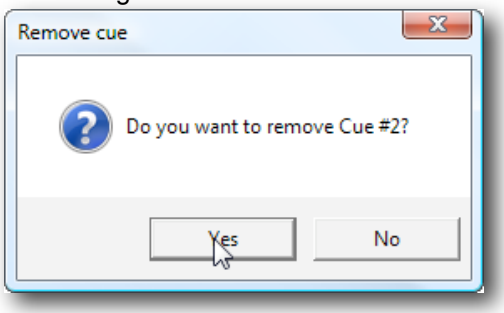
Deleting a cue

Theatre is a dynamic process. Throughout rehearsals scene shifts will be added, and scrapped as the performance is shaped. To keep up with the rapid changes, SpikeMark makes it just as easy to remove cues as it is to add them. Deleting a cue from the show is simple:

1. Right-click on the cue you delete in the **Cue Grid** and select **Delete Cue**.



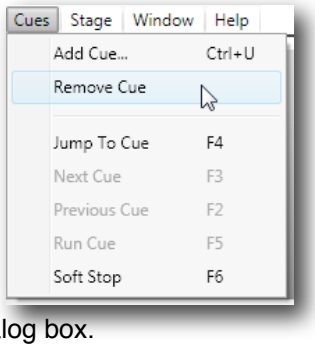
2. Click **Yes** in the confirmation dialog box.



3. There is no step three.

-or-

1. Select the cue you wish to delete in the **Cue Grid**.
2. From the menu select **Cues** -> **Remove Cue**



3. Click **Yes** in the confirmation dialog box.

Linking Cues

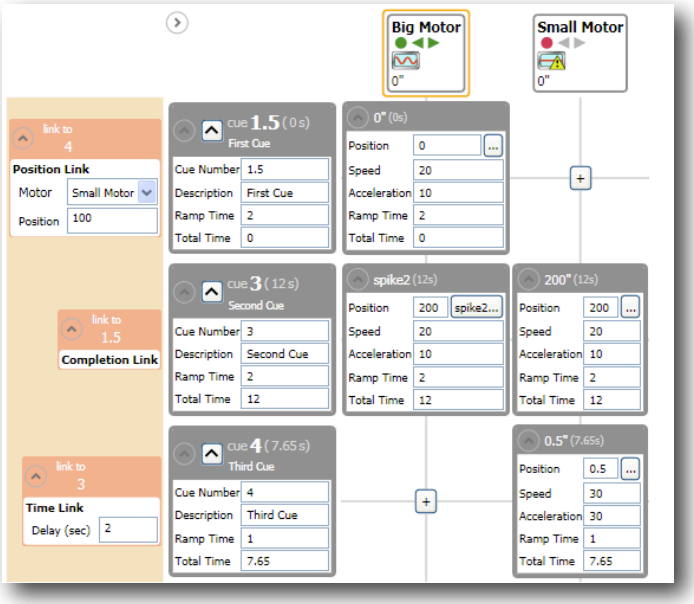
As you are writing cues in a new show, you'll find scene changes where simple cues fall short of the desired effect. You need to fire a sequence of movements, but they shouldn't all start at the same time. Maybe the movements should start with a 2 second delay between each. Perhaps the house wagon needs to wait to move upstage until the balcony unit is clear and in the wings. Whatever the need, links are the answer. Links empower you to create sequences of cues that start automatically in a staggered fashion.

As we discuss cue links, we need to have a vocabulary that identifies the cues that are involved in the sequence.

Parent Cue - The cue that is being watched by the link. The **Parent Cue** is run by either pressing the **GO** button, or it may be a **Child Cue** of another link in a longer sequence of linked cues. Whenever the parent cue is running, the link will check its criteria and automatically start the **Child Cue** when the criteria is met.

Child Cue - The cue that will be started by the link. When the link's criteria is satisfied, the link will automatically run the **Child Cue**. Graphically, child cues have the link displayed next to them in the **Cue Grid**.

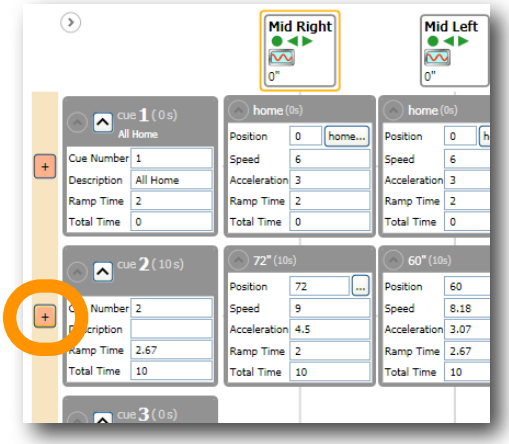
All links are displayed in the **Cue Grid** on the left margin.



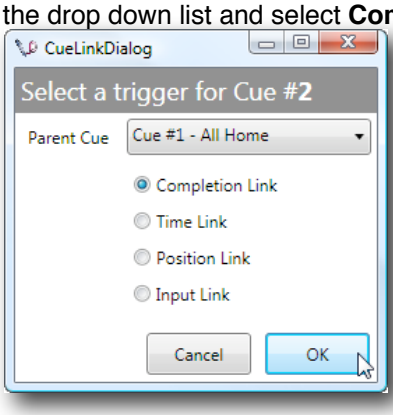
Linking by completion

You can link two cues together by completion so that whenever the Parent Cue successfully completes, the Child Cue will begin.

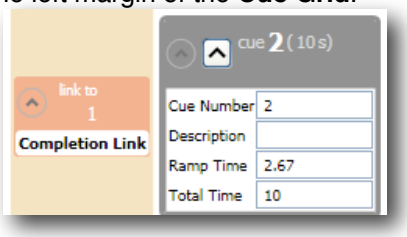
1. Click the **Add Link** button next to the **Child Cue**.



2. Select the **Parent Cue** from the drop down list and select **Completion Link**, then press **OK**.



3. The new link will appear in the left margin of the **Cue Grid**.

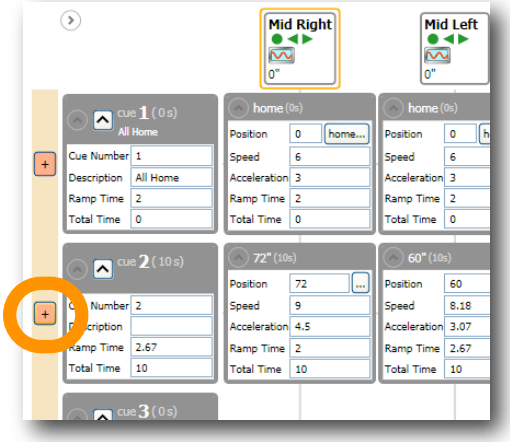


Note: You can create an infinite loop by creating an additional link between Cue 1 & Cue 2 where Cue 2 is the parent and Cue 1 is a child. This is handy when testing machines and in some shows where you need a motor to cycle back and forth indefinitely. To stop a loop of cues, you can press the Soft Stop button or Emergency Stop.

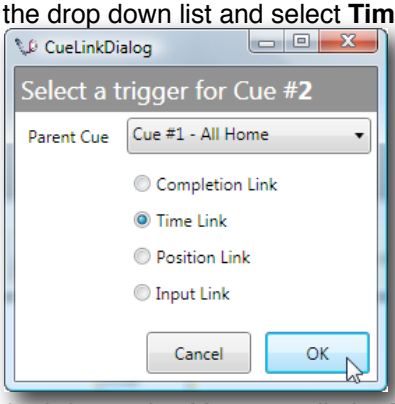
Linking by time

You can link two cues together with a time delay so that the **Child Cue** runs a programmed number seconds after the **Parent Cue** starts.

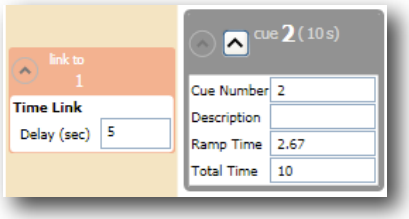
1. Click the **Add Link** button next to the **Child Cue**.



2. Select the **Parent Cue** from the drop down list and select **Time Link**, then press **OK**.



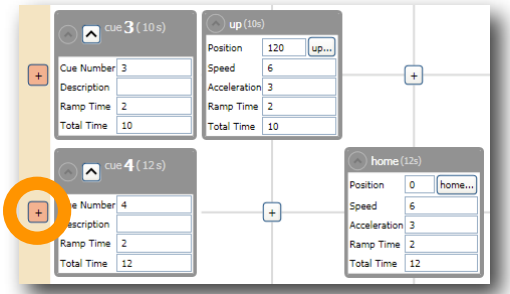
3. The new link is displayed in the left margin. You can edit the **Delay** value.



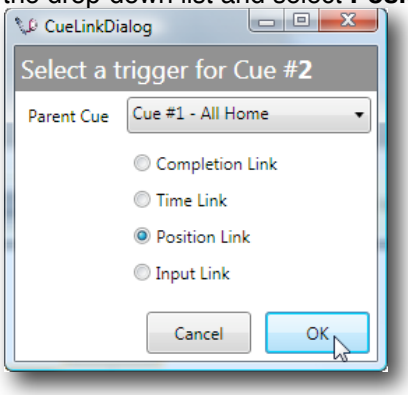
Linking by motor position

You can link a Child Cue so that it runs when one of the motors in the **Parent Cue** reaches a defined position. This is handy when you have to ensure that one motor has cleared the way for another motor.

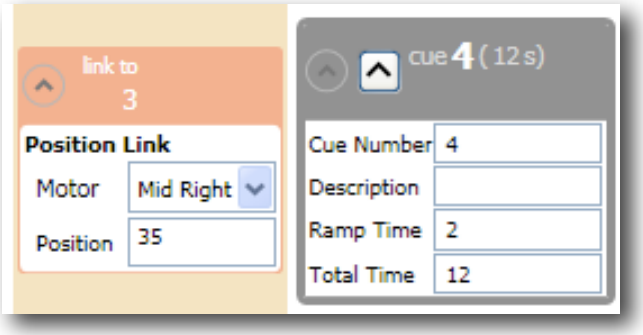
1. Click the **Add Link** button next to the **Child Cue**.



2. Select the **Parent Cue** from the drop-down list and select **Position Link**, then press **OK**.



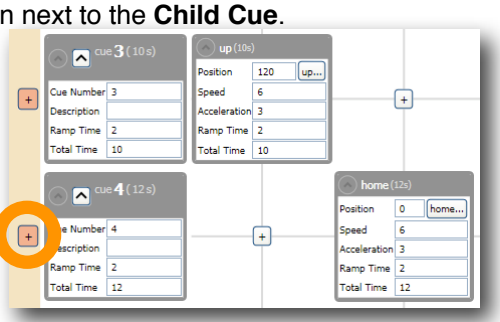
3. The new link will appear in the left margin next to the **Child Cue**. Select the motor from the **Parent Cue** and enter a position that will trigger the link.



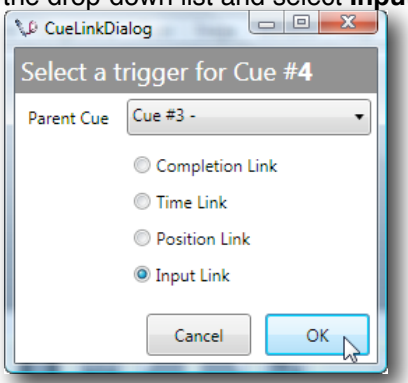
Linking by FX input

You can link a **Child Cue** to a **Parent Cue** so it runs as soon as an input switch is detected from a Stagehand FX. This can be handy if you have two cues that need to run in close proximity, but require a crew member to give a “Clear” signal for the second cue. In such a case, you could wire a pushbutton to the input of a Stagehand FX. When the **Parent Cue** runs, the crew member would push the button to start the **Child Cue** when appropriate.

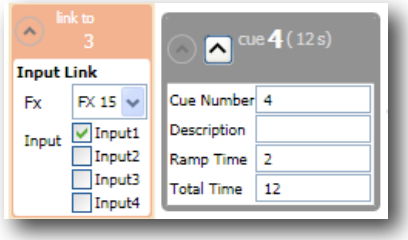
1. Click the **Add Link** button next to the **Child Cue**.



2. Select the **Parent Cue** from the drop-down list and select **Input Link**, then press **OK**.



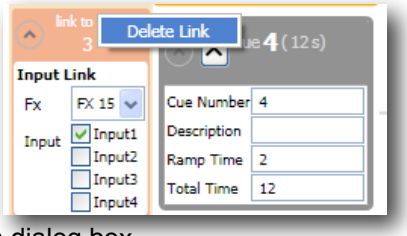
3. The new link will appear in the left margin of the **Cue Grid**. Select the Stagehand FX and the input that should trigger the **Child Cue**.



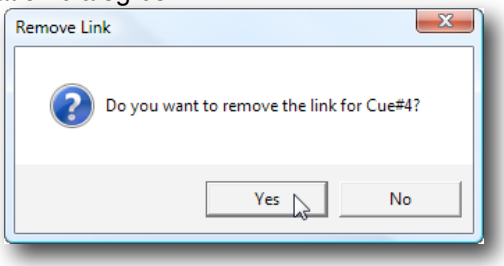
Delete a Link

Links are great for cue sequences that can be reliably run in the exact same way every performance. But if you find that a link is to rigid you may need to keep the cues independent and instead run the show with **Pile On** cues, which allows you to load and run multiple cues manually. To remove a **Link** from the show:

1. Right-click on the **Link** in the **Cue Grid** and select **Delete Link**.



2. Click **Yes** in the confirmation dialog box.



Using the Stage Model

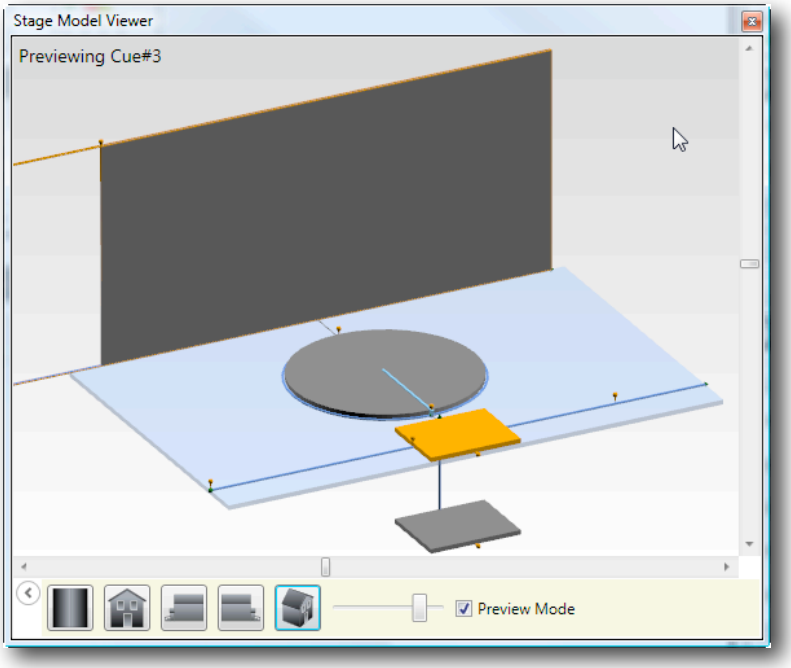
SpikeMark's **Stage Model** provides a cool way to visualize your show. Rather than seeing the bland rows and columns of numbers in the **Cue Grid**, the **Stage Model** shows you a three-dimensional schematic of your stage. Every motor in the show can be sized and positioned on the virtual stage so it closely resembles the actual scenery in your theatre. When you load a cue, any motor that is programmed to move will change color and display its destination allowing you to see at a glance what will move and where it will move to. As the motors move in real life, the move in the virtual model as well, giving you instant feedback about what is happening on the stage. This feedback is greatly appreciated when you are sitting back stage trying to keep track of every piece of moving scenery and every person on the stage. Looking at the **Stage Model** can make it much easier to quickly decide if someone is in harm's way before the Stage Manager shouts "GO!" in your ear.

Another great way to use the **Stage Model** is as a pre-tech cue-writing tool. Using **spikes** and the **Stage Model** you can rough-in your cues and clearly see what the stage will look like before setting foot in the theatre. This can give you and the artistic folks time to play with different looks before tech and really speed up the rehearsal process.

Let's take a look at the latter scenario, to discover how the **Stage Model** can be used when running shows see the next section Running Shows.

Navigating the Model

You can zoom, pan, and twirl the 3d model inside a **Stage Model Viewer**.

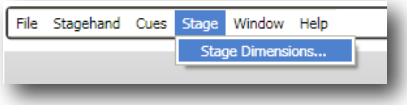


1. Zooming - The slider located next to the preset view buttons will let you zoom into the model. You can also use the scrollwheel on your mouse to zoom in or out.
2. Panning - Using the right mouse button, click and drag inside the 3d model to pan around the space.
3. Rotate - Use the scrollbars on the right and bottom of the model viewer to orbit around the stage.
4. Preset Views - Use the view buttons to quickly jump to the 5 orthographic projections (plan, front, right, left, & isometric).

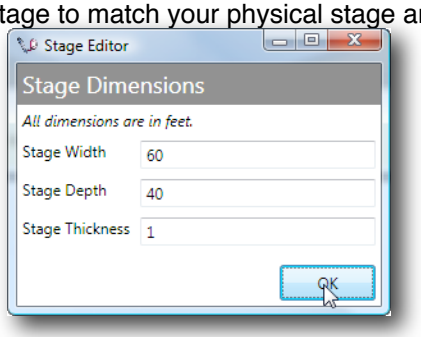
Setting the Stage Dimensions

In the **Stage Model**, the stage is shown as a semi-transparent box. To make the **Stage Model** more closely resemble the size of your stage, you can change the default stage dimensions.

1. From the menu bar, select **Stage -> Stage Dimensions...**



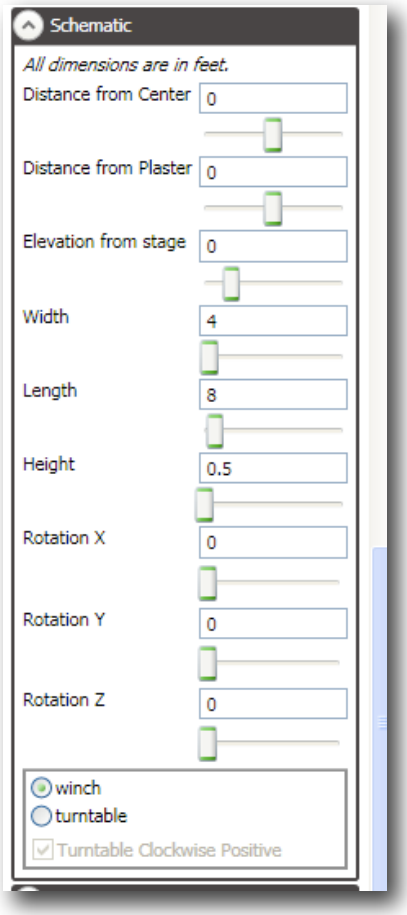
2. Edit the dimensions of the stage to match your physical stage and click **OK**.



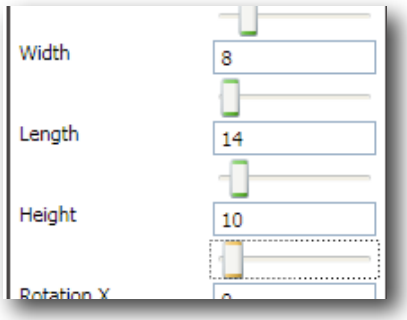
Make a Winch Model

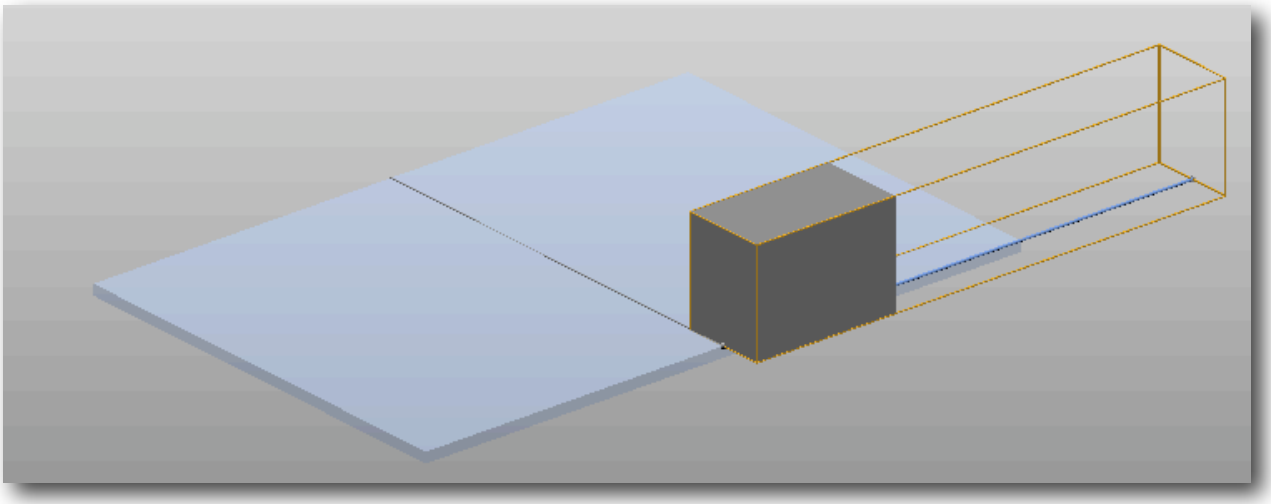
After you've changed the size of the stage to match your theatre space, the next step is to make the motors look roughly like the scenery on the stage. Each motor in the show can be viewed as either a winch or turntable. Winches always look like a box traveling along a track. The length of the track is determined by the **Max Forward** and **Min Reverse** parameters. The size and placement of the scenery on the track can be adjusted in the **Stagehand Editor** also.

1. Select the motor you want to adjust in the **Cue Grid**.
2. In the **Stagehand Editor** scroll until you find the **Schematic** section.

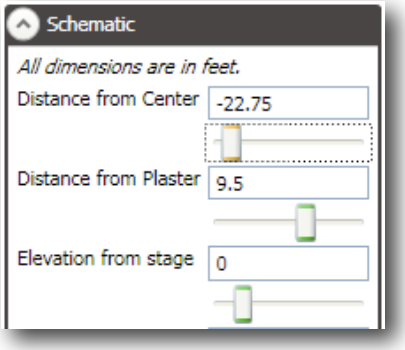


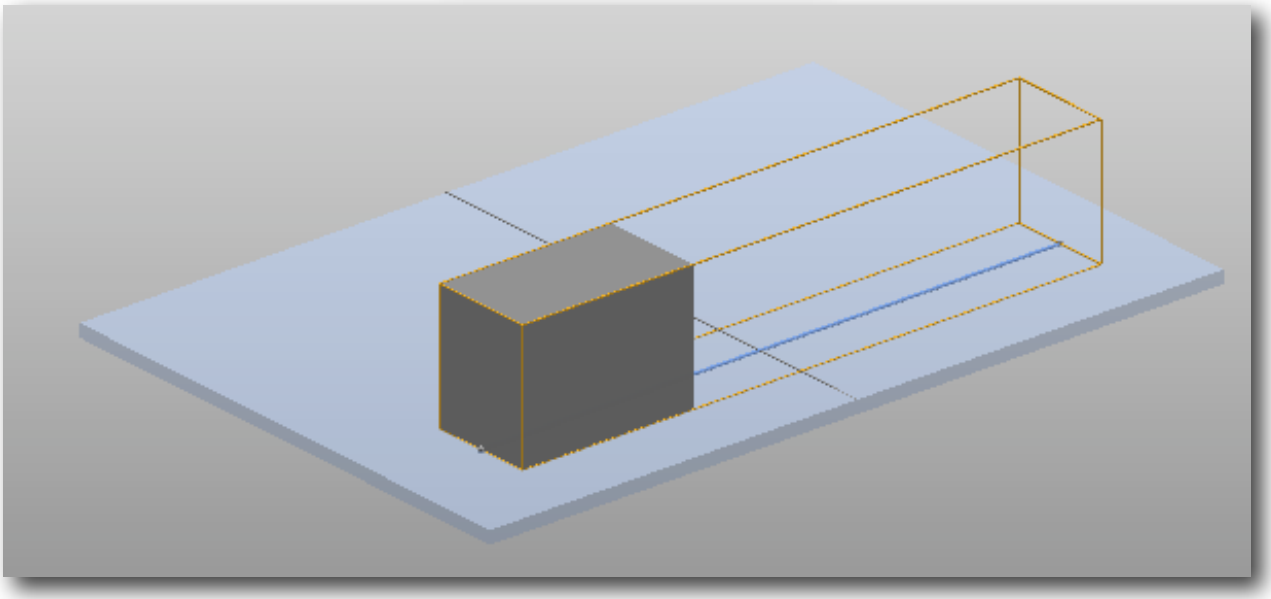
3. Make sure that winch is selected at the bottom of the list. (We'll look at turntables later).
4. Adjust the **Width**, **Length**, and **Height** properties by using either the sliders beneath each text field, or typing in a dimension directly. As you change the dimensions, the winch model in the **Stage Model Viewer** will update. Note that all dimensions are given in feet.



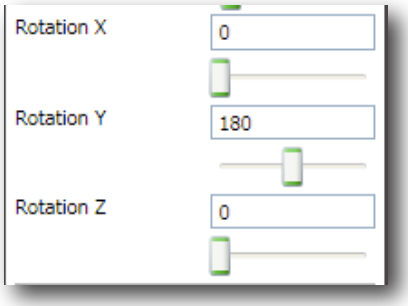


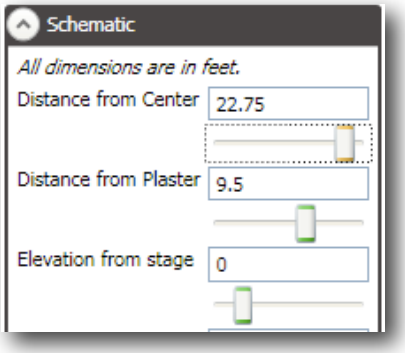
5. Now, move the winch into position so that it matches your stage. Use the **Distance from Center**, **Distance from Plaster**, and **Elevation from stage** sliders to get the position close. You can then fine tune the position by typing in accurate dimensions from your groundplan.

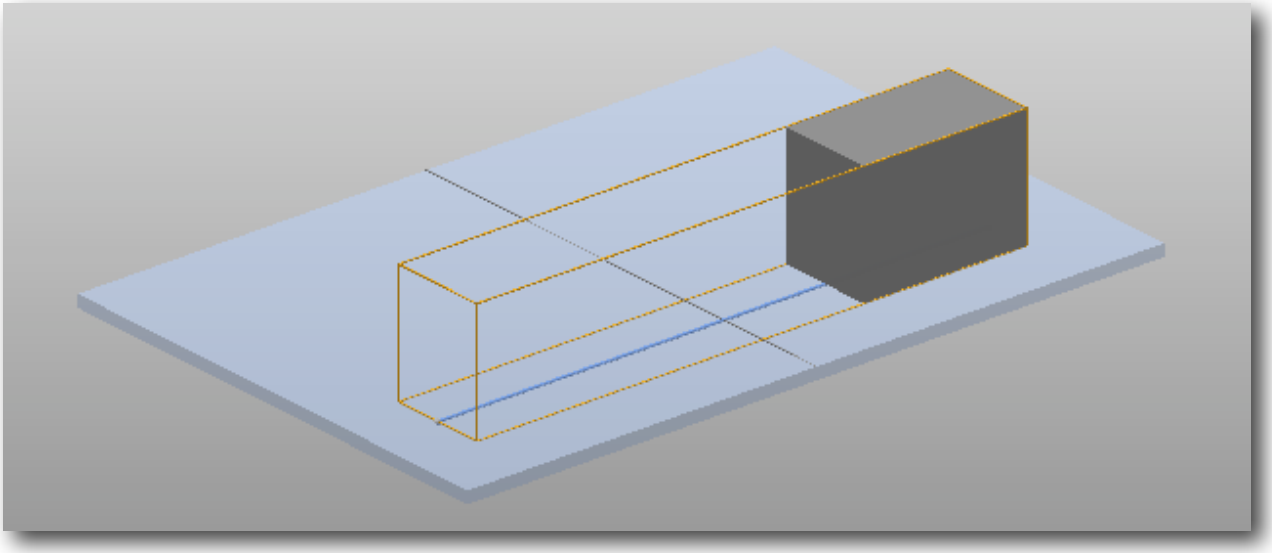




6. By default, a winch is displayed with its **Min Rev Position** on stage right and the forward direction is heading stage left. You can use the **Rotation Y** parameter to spin the winch around and then reposition it if you need the **Min Rev Position** to be on stage left.



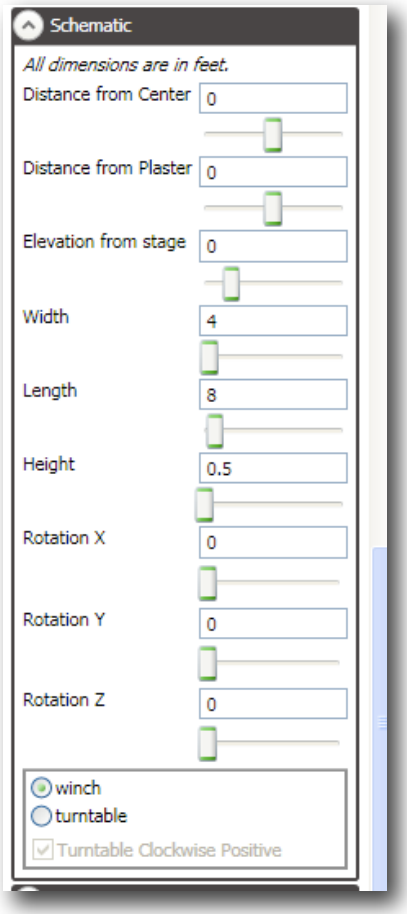




Make a Lift Model

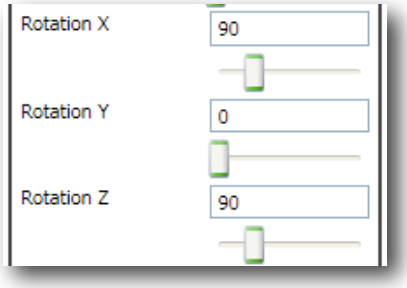
The winch-style model can be used to represent various machines other than ordinary deck winches. By adjusting the size and rotation of the winch we can easily make it look like a lift.

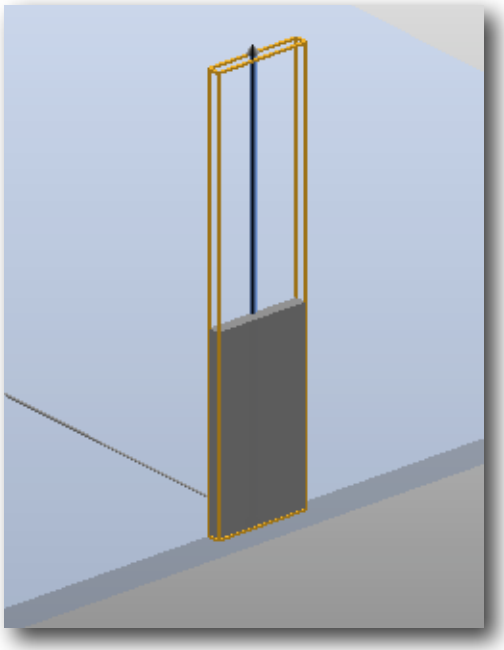
1. Select the motor you want to adjust in the **Cue Grid**.
2. In the **Stagehand Editor** scroll until you find the **Schematic** section.



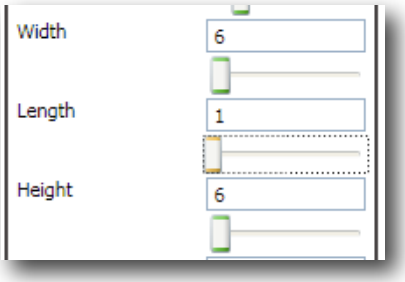
3. Make sure that winch is selected at the bottom of the list. (We'll look at turntables later).

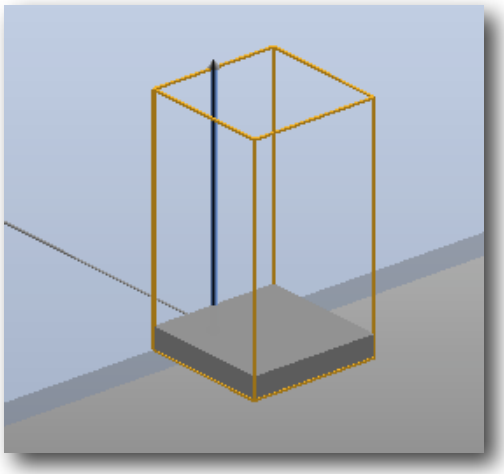
4. Using the **Rotation Z** and **Rotation X** sliders, rotate the winch model until it is standing upright.



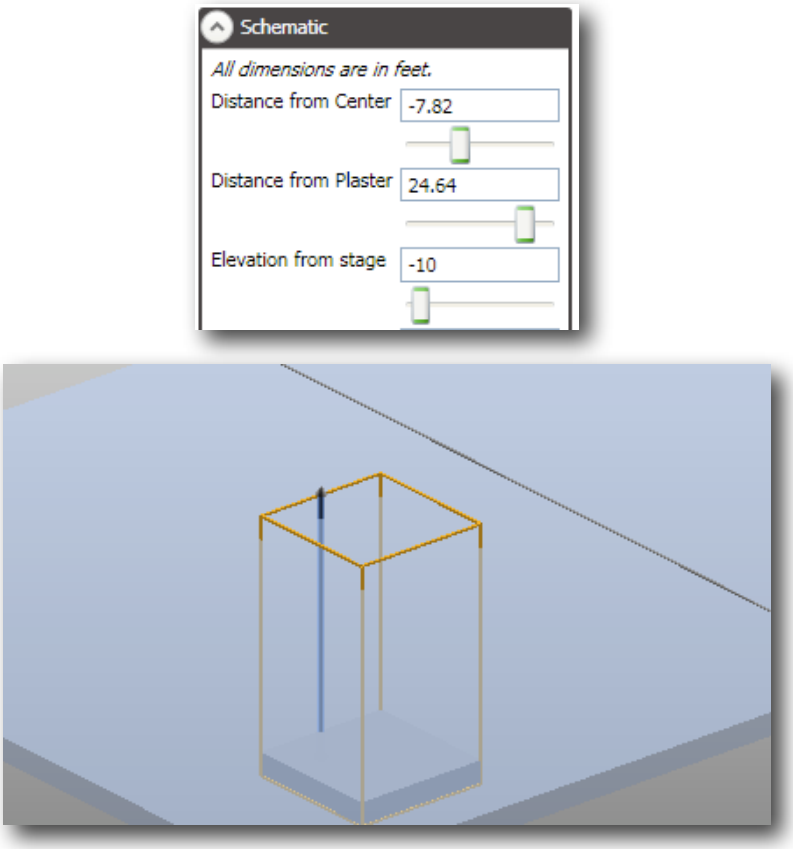


5. Now use the **Width**, **Length**, and **Height** sliders to make the winch look like a lift platform.





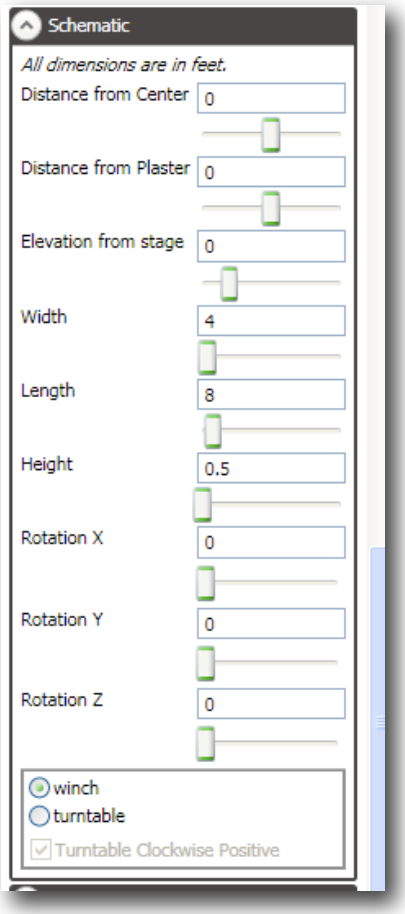
6. Using the **Distance from Center**, **Distance from Plaster**, and **Elevation from stage**, position the lift upstage right and beneath the stage floor so that it travels into the trap room. The stage is semi-transparent so that you can see scenery moving even if it is beneath the stage.



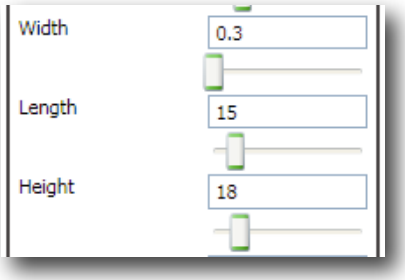
Make a Curtain Model

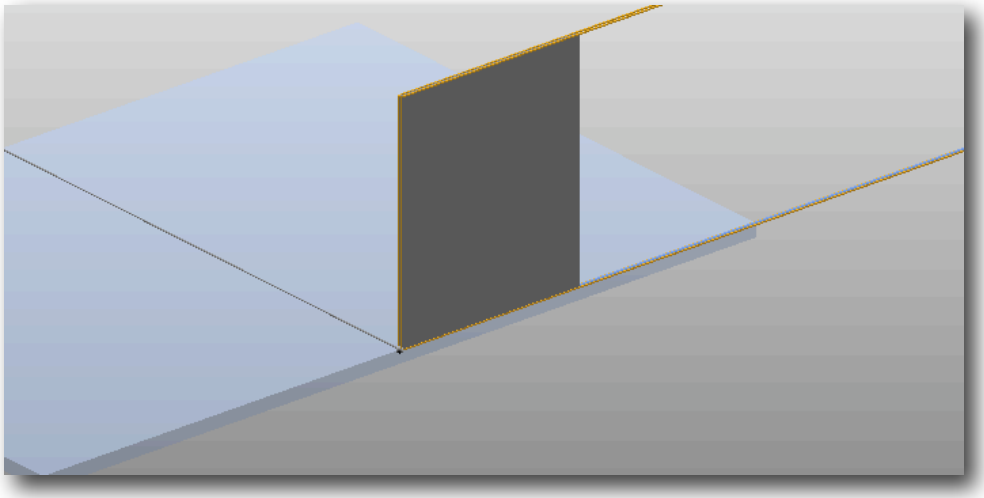
The winch-style model can be used to represent various machines other than ordinary deck winches. By adjusting the size and rotation of the winch we can easily make it look like either a traveling curtain or a flying curtain.

1. Select the motor you want to adjust in the **Cue Grid**.
2. In the **Stagehand Editor** scroll until you find the **Schematic** section.

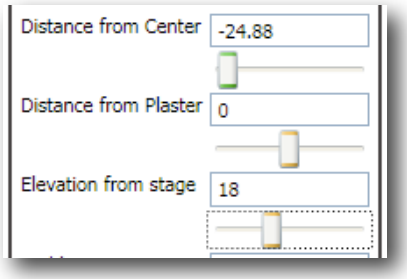


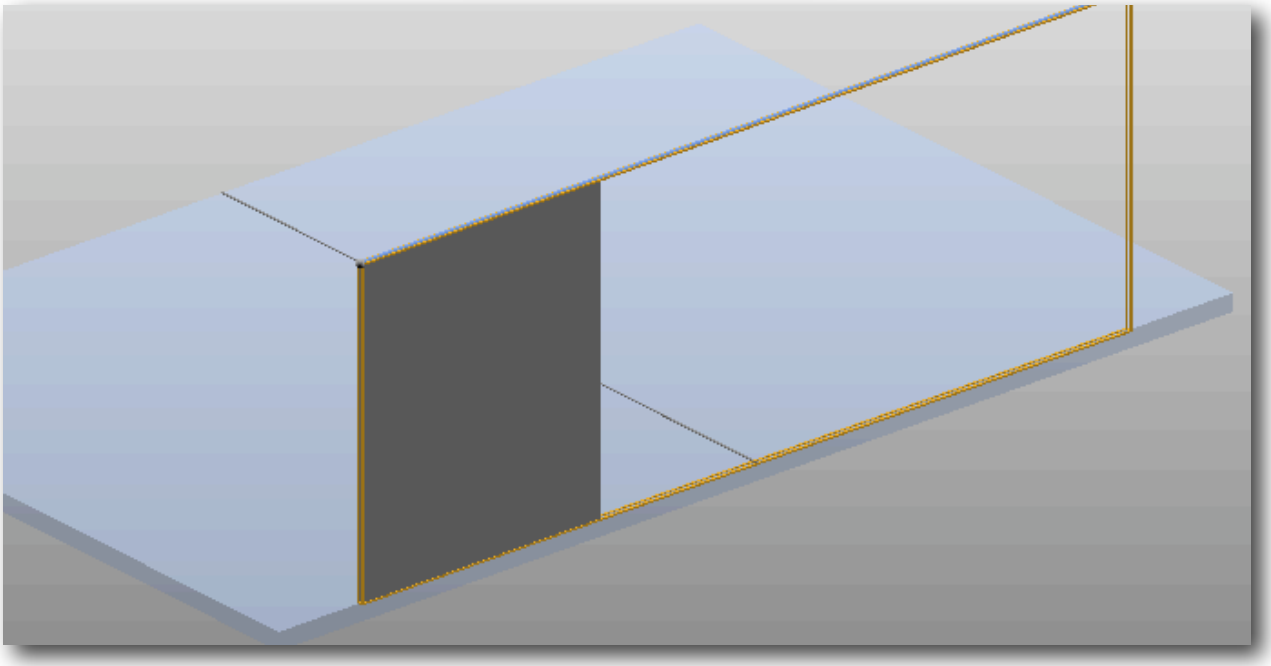
3. Make sure that winch is selected at the bottom of the list. (We'll look at turntables later).
4. Adjust the **Width**, **Length**, and **Height** properties to make a tall, thin curtain.



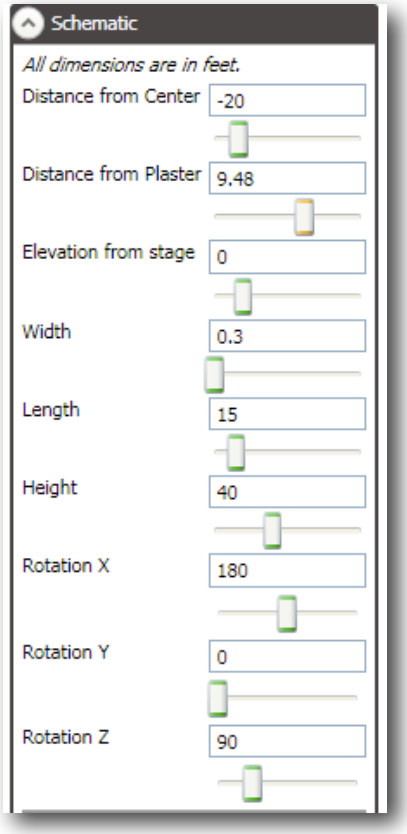


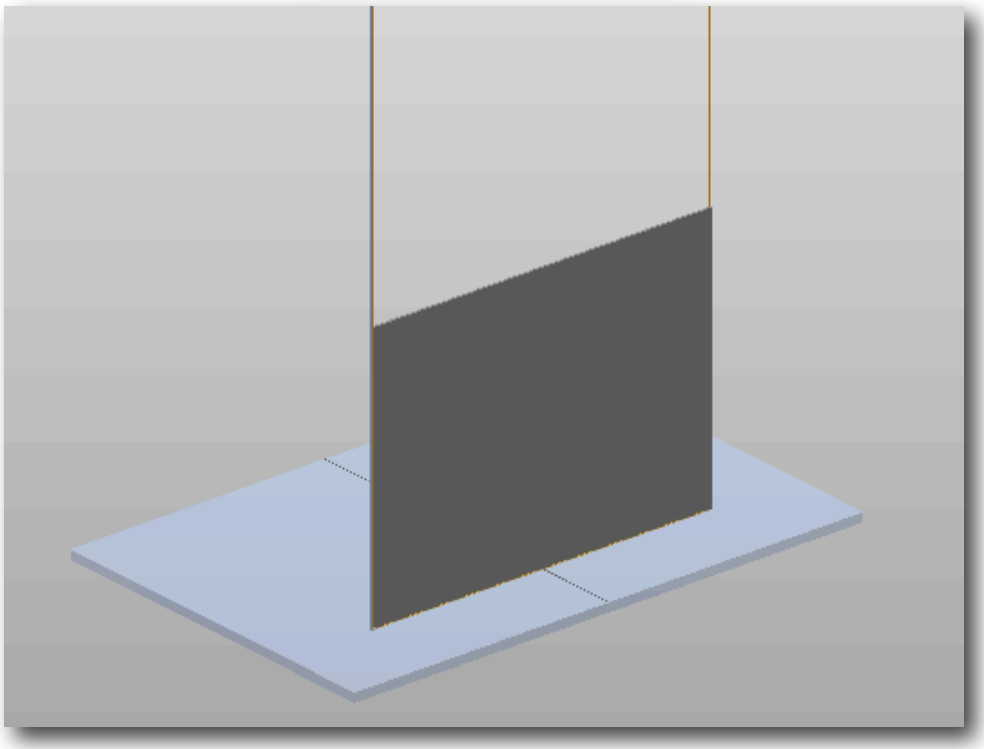
5. Position the curtain on the stage.





6. By adjusting the position, size, and rotation, you can change the horizontal traveler curtain into a vertical flying curtain.

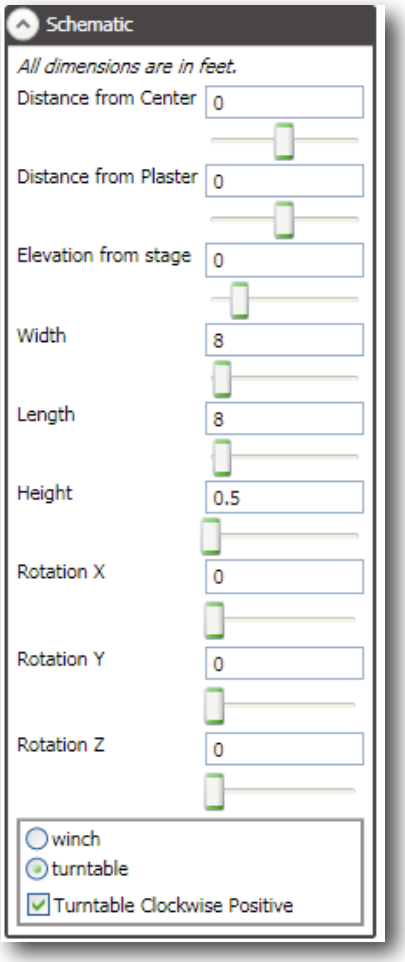


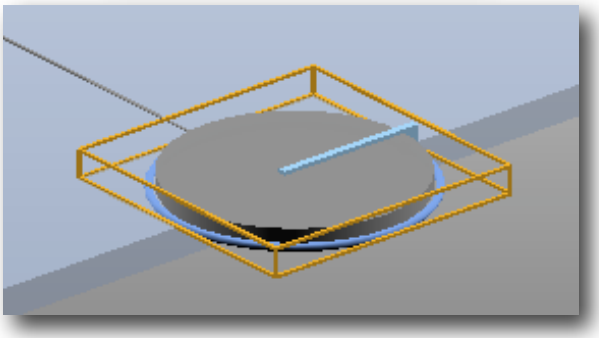


Make a Turntable Model

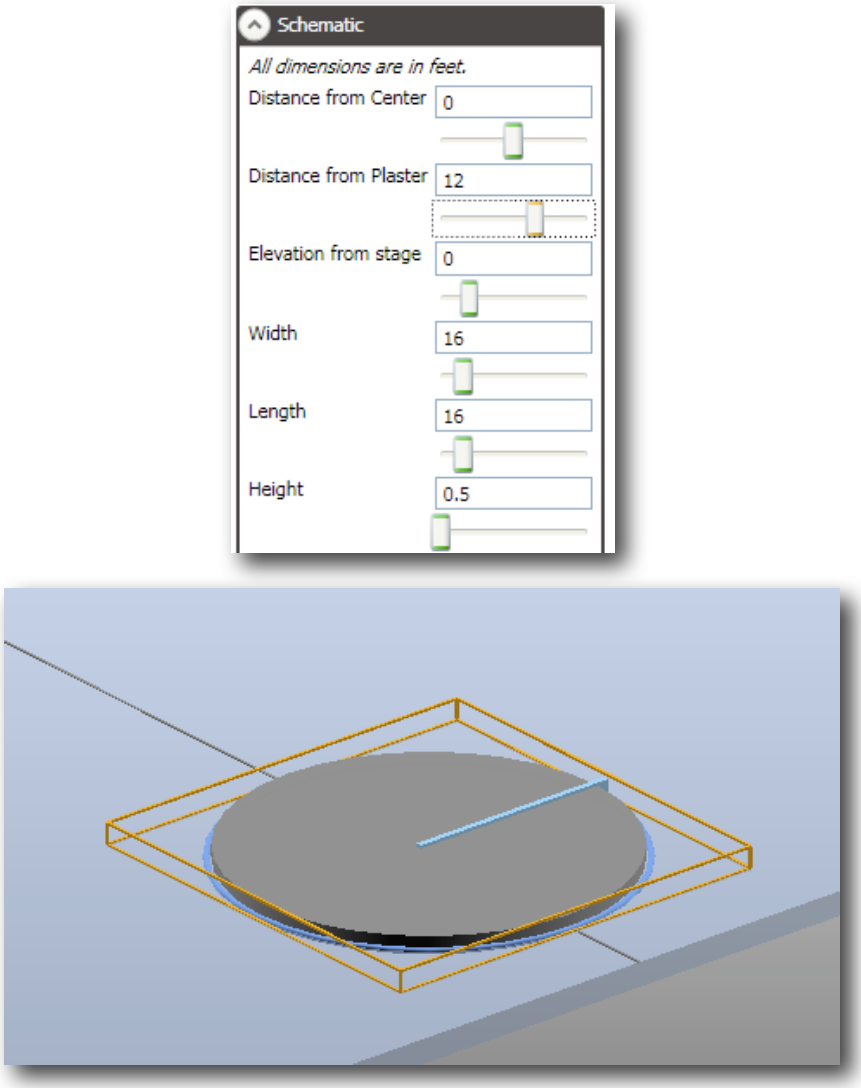
The default model style for every motor in the show is a winch which move linearly along a track. This works great for deck winches, lifts, and curtains which all move in a linear fashion. However, a lot of shows include turntable so SpikeMark makes it easy to change a motor model from winch to turntable.

1. Select the motor you want to adjust in the **Cue Grid**.
2. In the **Stagehand Editor** scroll until you find the **Schematic** section.
3. Select **turntable** instead of **winch** at the bottom of the **Schematic** section.

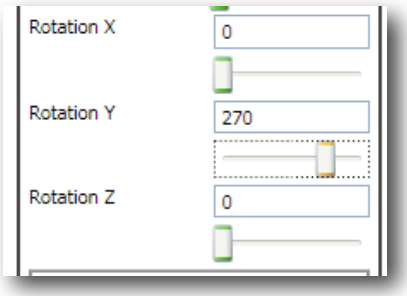


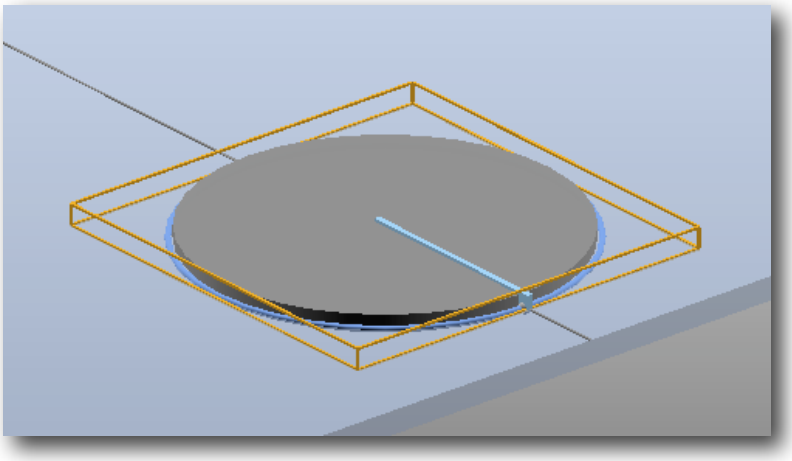


4. Use the **Distance from Plaster** and **Width** sliders to position and resize the turntable.



5. By default, the turntable model orients position zero (“0 degrees”) towards stage left. You can change this orientation by using the **Rotation Y** slider.



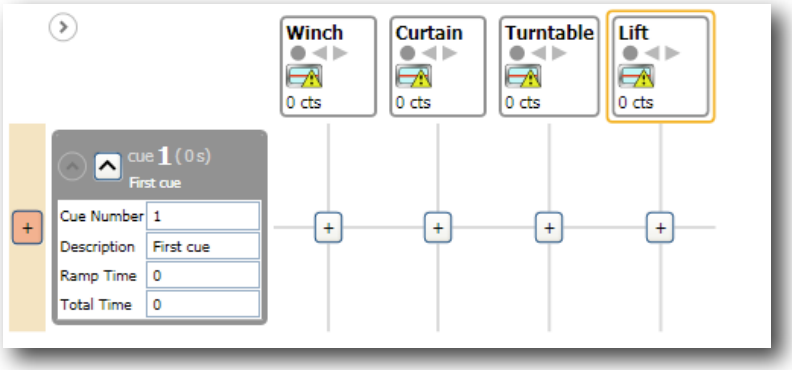


Writing Cues in the Stage Model

In this section we're going to walk through the process of writing a couple of cues for a show without being connected to any motors. You might want to do this as part of a paper tech to get some of the rough cue writing done before the show is loaded in. This way you can sit around a computer with your artistic staff and agree on some rough looks and basic transition timings to save time during tech.

To get started you can add some motors to a show and fill in only the properties needed to write the cues, leaving configuration details like **Position Scale** and **IP Addresses** to be confirmed and entered once you are on stage and can work with the real equipment.

1. With SpikeMark started. Add four motors and rename them as shown.



2. Edit the parameters for Winch as listed.

Parameter	Value
Max Fwd Position	540
Position Units	inches
Max Speed	30
Default Speed	20
Default Accel	10
Distance from Center	-27.5

Parameter	Value
Distance from Plaster	4
Width	8
Length	10

3. Edit the parameters for Curtain as listed.

Parameter	Value
Max Fwd Position	600
Position Units	inches
Max Speed	30
Default Speed	28
Default Accel	14
Distance from Center	-74
Distance from Plaster	35
Width	0.3
Length	50
Height	24

4. Edit the parameters for Turntable as listed.

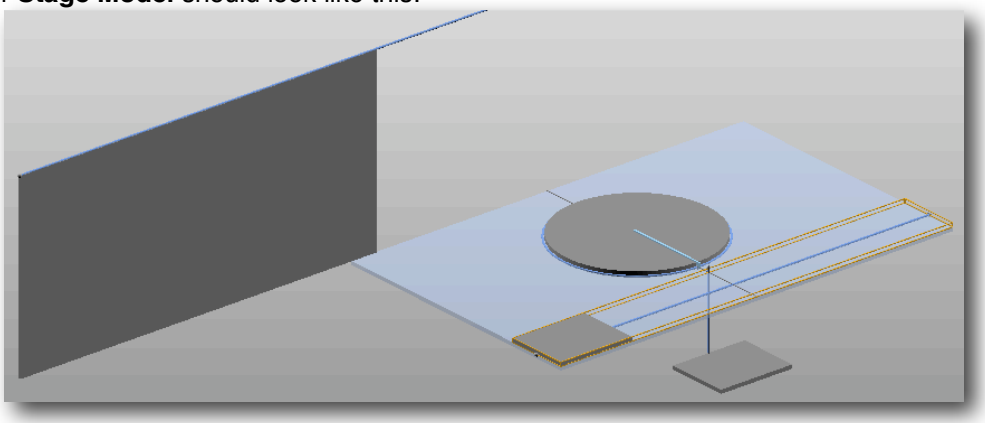
Parameter	Value
Max Fwd Position	720
Min Rev Position	-720
Position Units	degrees
Max Speed	18
Default Speed	6
Default Accel	3
Distance from Center	0
Distance from Plaster	20
Elevation from stage	0
Width	20

Parameter	Value
Height	0.5
Rotation Y	270

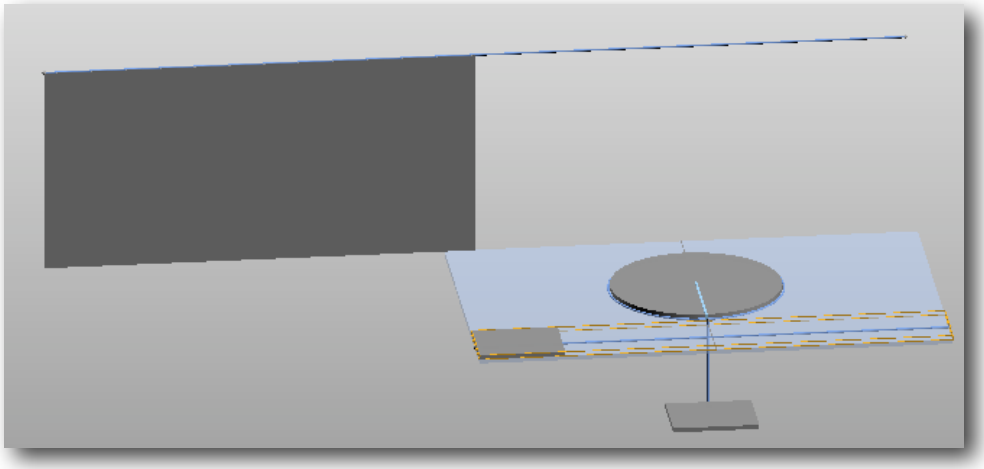
5. Edit the parameters for Lift as listed.

Parameter	Value
Max Fwd Position	120
Position Units	inches
Max Speed	20
Default Speed	15
Default Accel	7
Distance from Plaster	8
Elevation from stage	-10
Width	10
Length	0.5
Height	8
Rotation X	90
Rotation Z	90

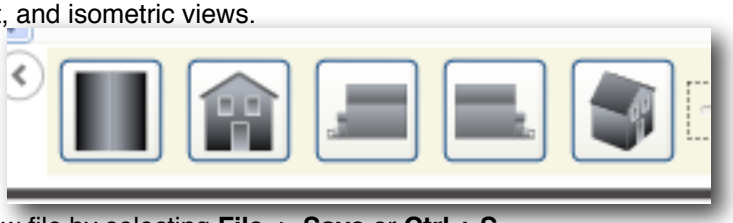
6. Your **Stage Model** should look like this:



7. Using the vertical scrollbar on the right side, and the horizontal scrollbar along the bottom of the **Stage Model Viewer**, rotate the view a bit to make it easier to see all the scenery.



8. You can also use the **View** buttons in the **Stage Model Viewer** to easily switch between top, front, right, left, and isometric views.



9. Save your show file by selecting **File -> Save** or **Ctrl + S**.

We have four motors setup on the virtual stage. Before we start writing cues, we need to add some spikes to each motor to identify the positions we'll likely use in the cues. Don't worry about getting the position values exactly right in the spikes, they can be edited at any time and will automatically update in all your cues.

1. Add the following spikes to Winch.

Spike	Value
SR	0
Center	270
SL	540

2. Add the following spikes to Curtain.

Spike	Value
Off	0
On	600

3. Add the following spikes to Turntable.

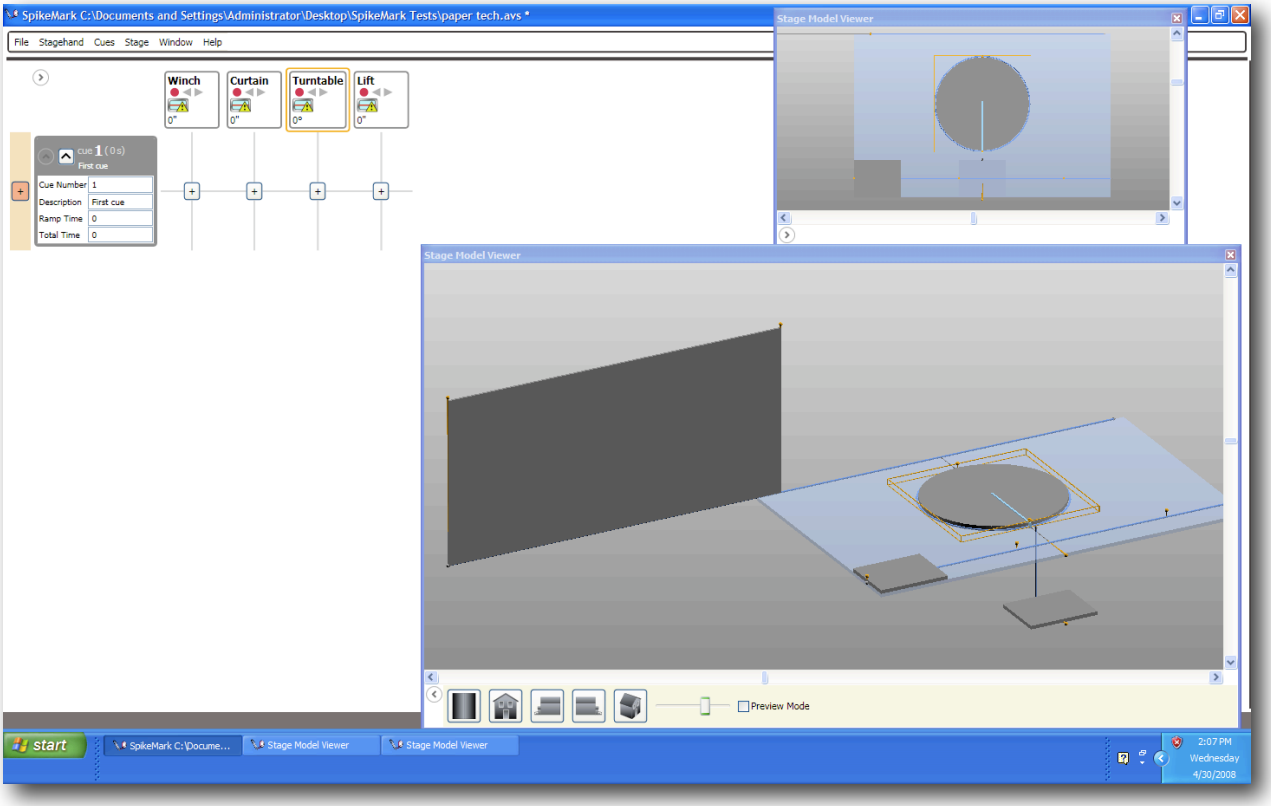
Spike	Value
DS	0
US	180

4. Add the following spikes to Lift.

Spike	Value
Down	0
Up	120

5. Save your show.

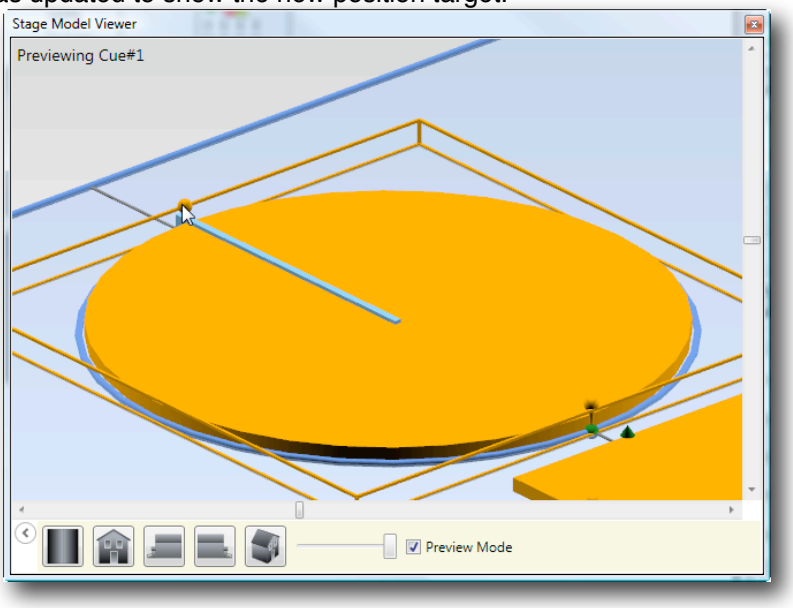
With the spikes added, I like to reconfigure the interface to make it more friendly for paper tech. Expand the **Cue Grid** to take up the whole screen and then add a couple of **Stage Model Viewers** by selecting **Window -> Stage Model Viewer** twice. Resize and arrange to your liking. Remember that you can navigate inside the **Stage Model Viewer** by panning (right mouse drag), zooming (mouse scrollwheel), and rotating (scrollbars on the right and bottom). Here's what my workspace looks like.



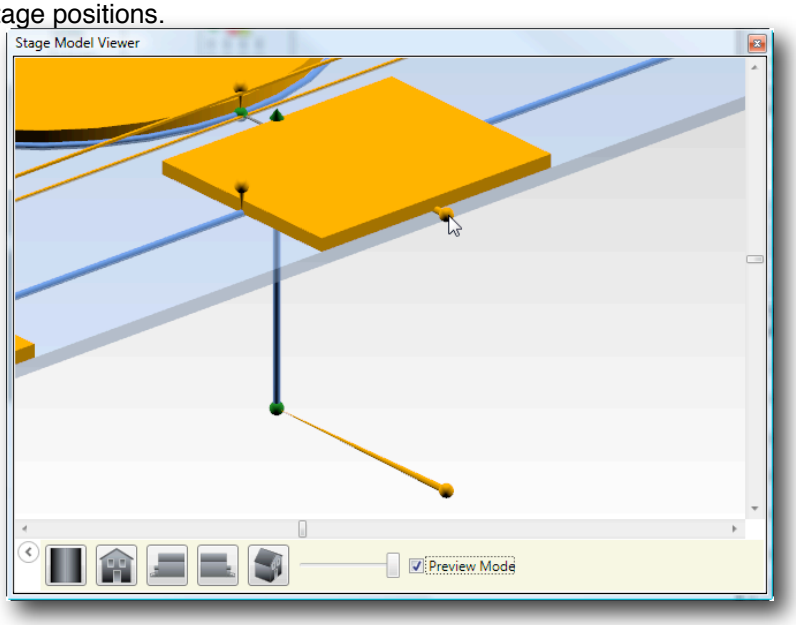
All right, let's write add some motors into Cue 1. In this cue I want to spin the Turntable upstage, raise the Lift, drive the Winch and Curtain offstage.

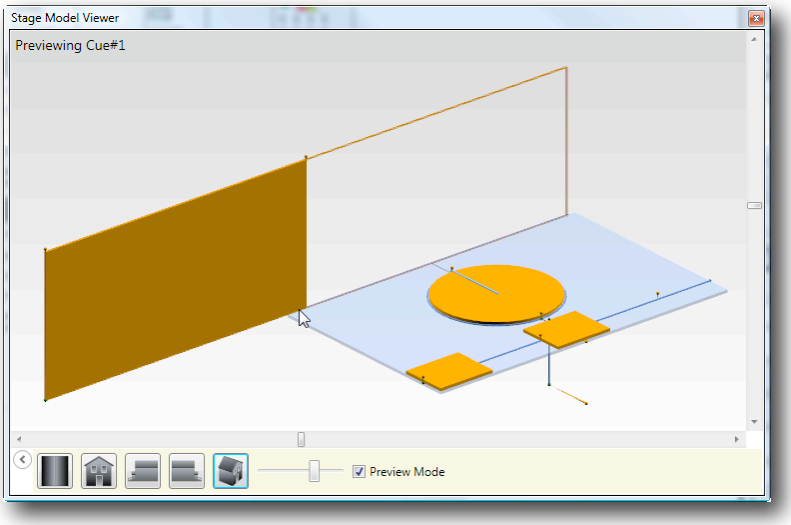
1. Add all the motors into Cue 1 by using the **Add Movement** buttons in the **Cue Grid**.
2. Click on Cue 1 in the **Cue Grid** to select it.
3. Check the **Preview Mode** box on the **Stage Model Viewer**. The motor's that are included in Cue 1 should turn orange. The **Stage Model View** background changes and indicates what cue is being previewed in the upper left corner.
4. We need to set the positions of all the motors. To change the target positions, we can click on the appropriate spike icons in the model. Zoom in and pan around (using the scrollwheel and right

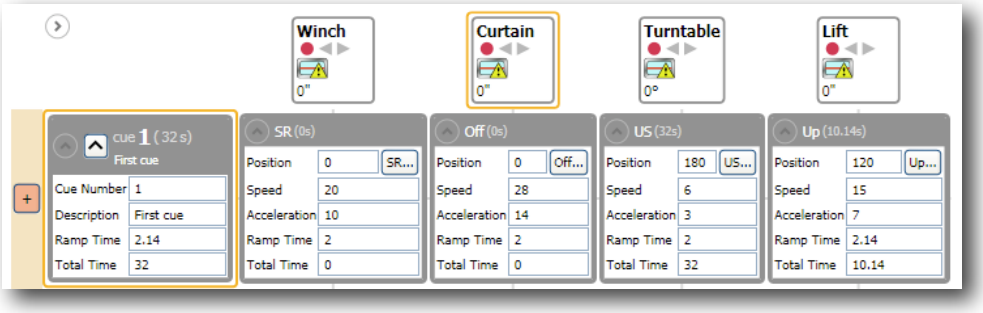
mouse button) until you can locate the upstage spike for the Turntable and click it. Notice that the **Cue Grid** has updated to show the new position target.



5. Repeat the process to change the Lift position to its upper spike and both the Winch and Curtain to their offstage positions.

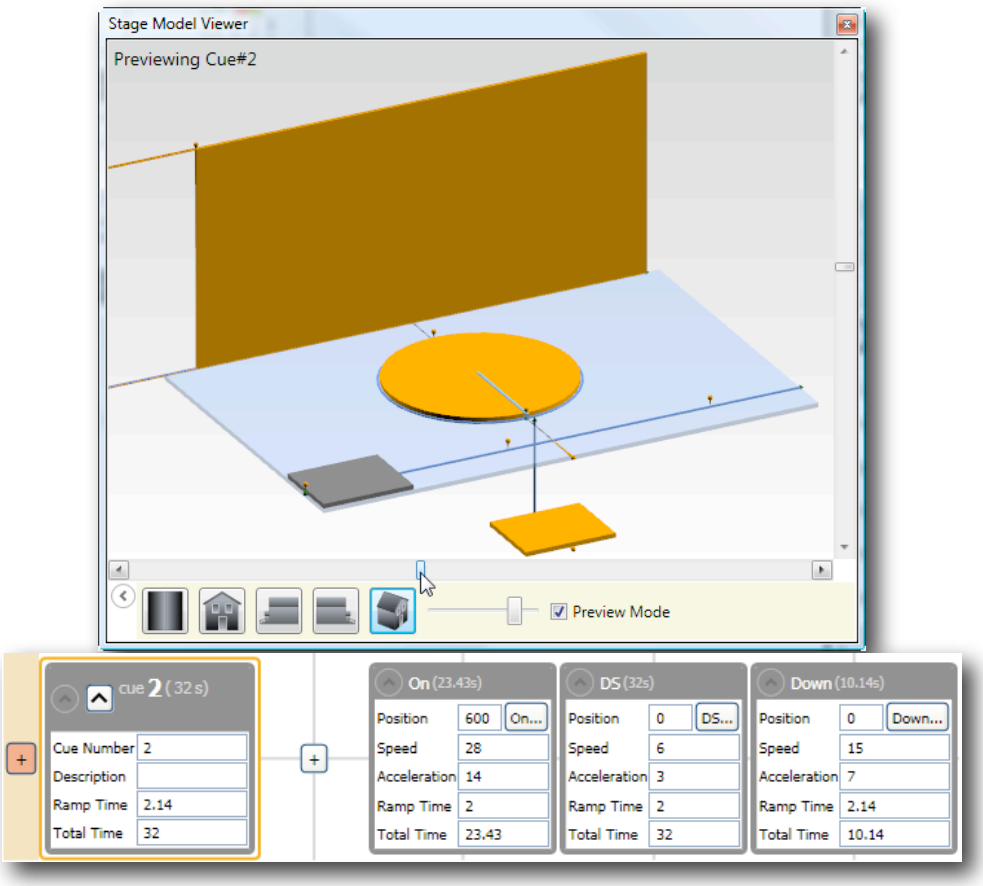






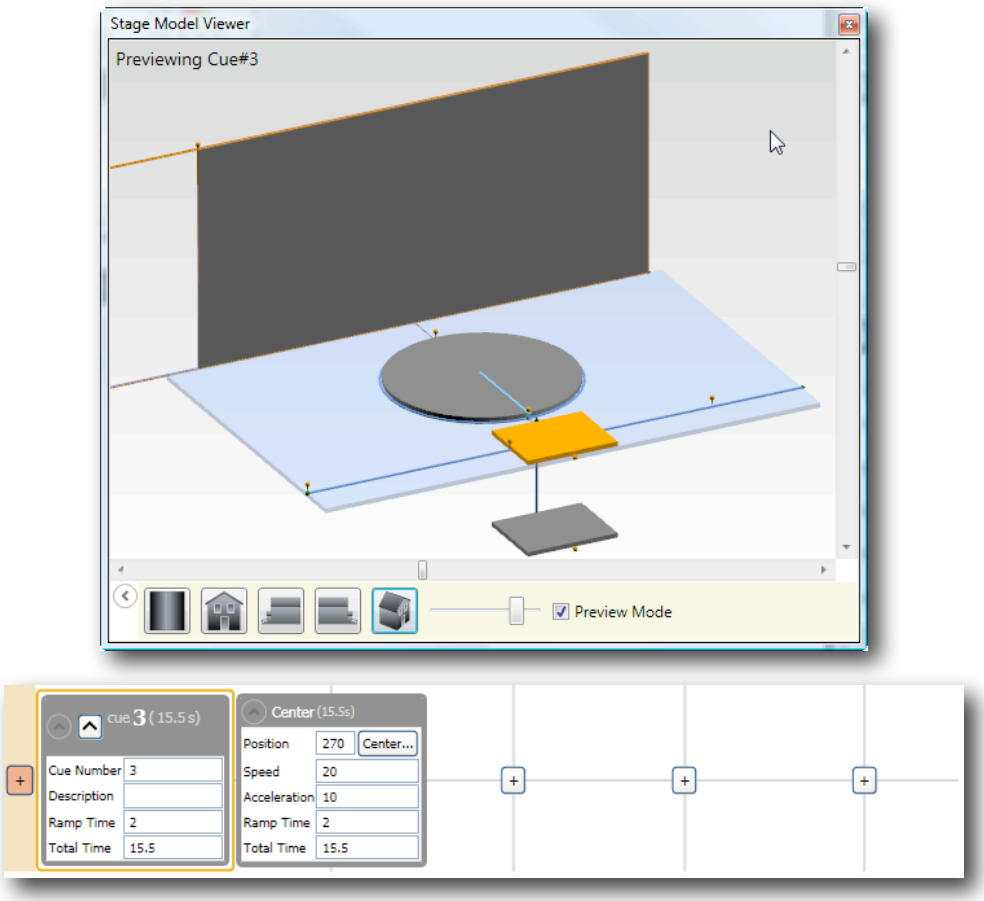
For our second cue, let's drop the Lift, bring the Curtain onstage, and spin the Turntable back downstage.

1. Add a second cue in the **Cue Grid**.
2. Add movements for Lift, Curtain, and Turntable.
3. Select Cue 2 in the **Cue Grid**, and confirm that the **Stage Model Viewer** displays "Previewing Cue #2".
4. Set spikes for the three motors so that Cue 2 looks like this:



Add a Cue 3 that drives the winch to center to cover the hole in the stage that was likely left after the lift went to the basement. Once you get to rehearsal, you may want to link Cue 3 to Cue 2 so that the winch starts to move as soon as the Lift is clear of the Winch path.

1. Add a Cue 3 in the **Cue Grid**.
2. Add a movement for Winch in Cue 3.
3. Click on the center stage spike for the Winch, so that Cue 3 looks like this:



It's worth pointing out that when we preview Cue 3, the **Stage Model** displays the Curtain, Turntable, and Lift sitting in their Cue 2 position. The motors that aren't moving in Cue 3 are shown in gray at the position of their last cue. This gives you an easy way to visualize where all the motors in the show will be at any given point in the show, regardless of whether the motor is moving in the previewed cue. Take a moment and preview all the cues by selecting them in the **Cue Grid** and watching the **Stage Model Viewer** update.

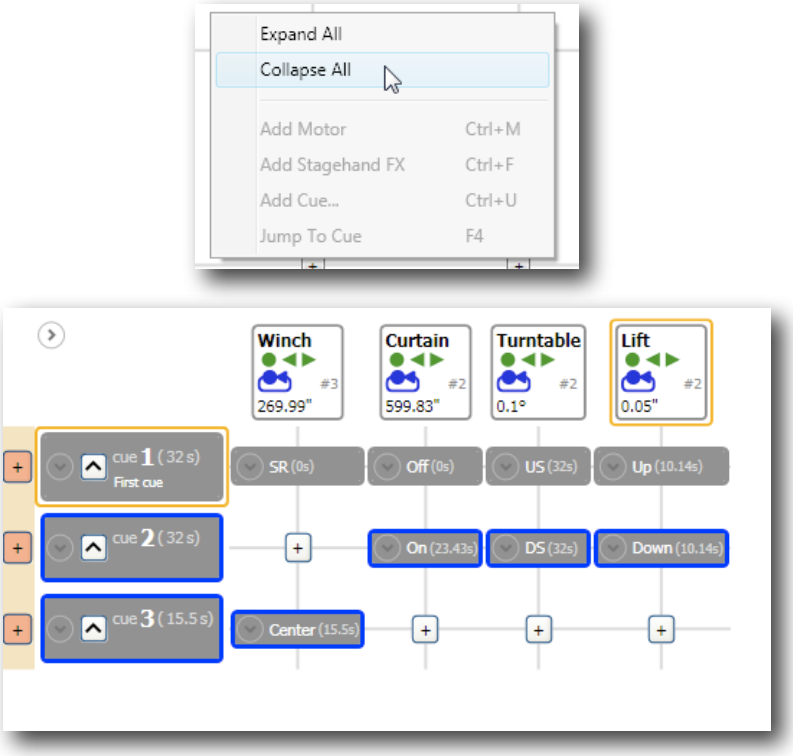
Running Shows

All of the setup, configuring, and cue writing is, in the end, just a prerequisite for what you really need to do -- run a performance. As the balance of writing in this manual may indicate, running a show is actually the easiest part of the process. Thankfully, by doing all your preparation attentively, running shows is easy. We'll look at a few ways to execute cues, navigate through the show, and explain what to do when something doesn't work as you expected.

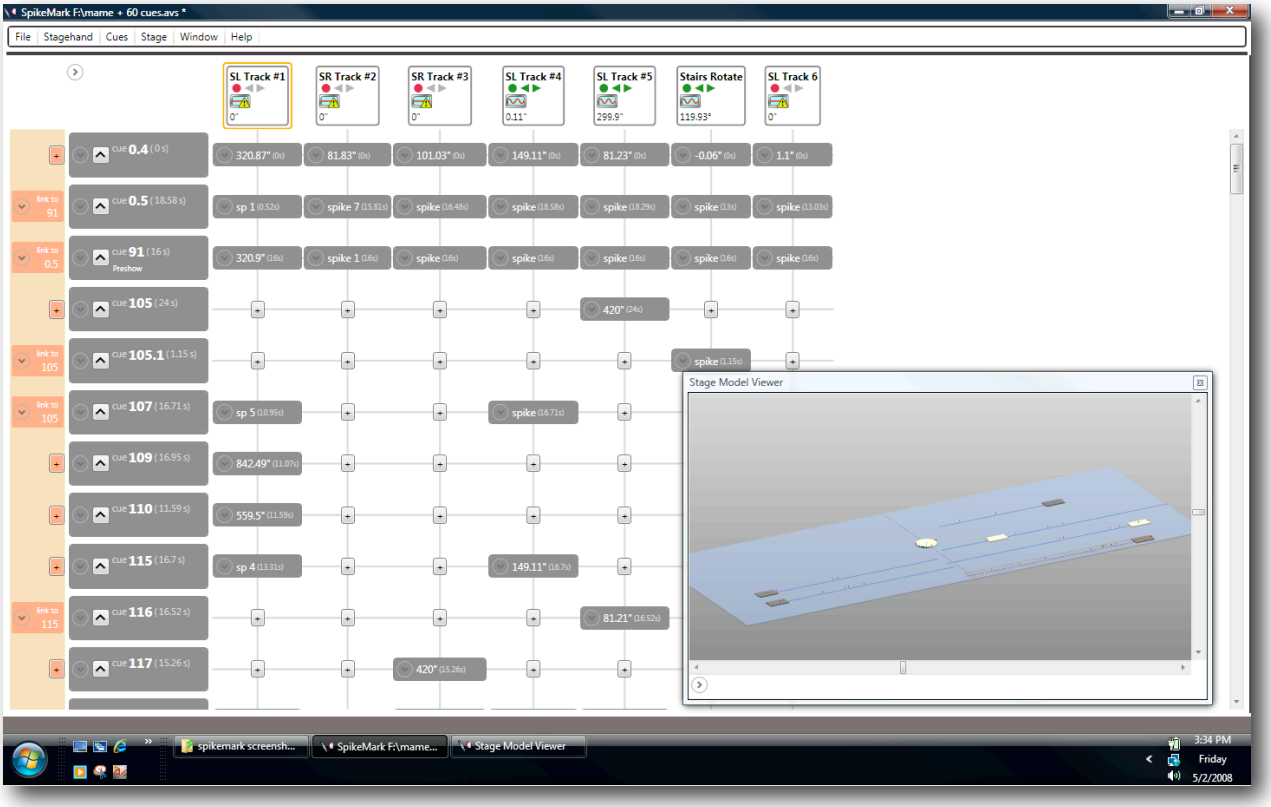
Setting up the Workspace

Once you've finished writing and editing cues, there are a few things you may want to arrange differently in the interface to see the appropriate amount of information. The way in which you like to set up the user interface is a personal preference, so the points outlined here are merely suggestions.

1. Expand the **Cue Grid** to it's full size by collapsing the **Stage Model Viewer** top pane, and the **Stagehand Editor** right-hand pane.
2. Hide all the speed and acceleration information for the cues by right-clicking on the **Cue Grid** and selecting **Collapse All**. This will compact the **Cue Grid** while allowing you to see the time each cue will take and the positions of every motor.



3. Create a floating **Stage Model viewer** that you can position in some unoccupied space of the **Cue Grid** or drag over to a second monitor.


SpikeMark F:\mame + 60 cues.avs *
File | Stagehand | Cues | Stage | Window | Help
SL Track #1
SR Track #2
SR Track #3
SL Track #4
SL Track #5
Stairs Rotate
SL Track 6
Stage Model Viewer

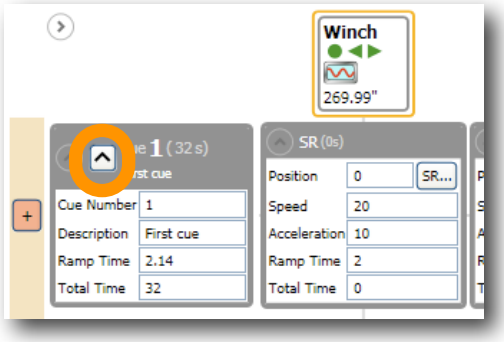
Loading a Cue

Executing a cue is a two step process. The first step is to load the cue. Loading the cue sends all the cue information out to the Stagehands and gets them ready to move. Typically, you'll load a cue when the stage manager gives you a "warning" for the cue.

Note: Cues that are linked do not need to be explicitly loaded, the link will automatically load and execute its child cue.

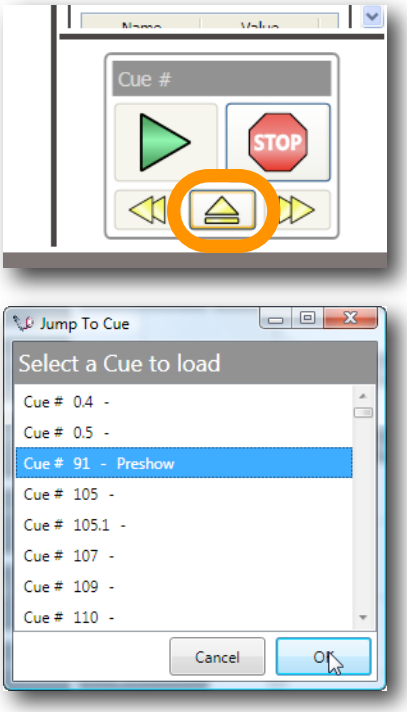
There are a few ways to load a cue, you can use whichever is most convenient:

1. Click the **Load Cue** button in the **Cue Grid**.



-or-

2. Click the **Jump To Cue** button on the **Cue Controller** and select a cue from the dialog box that appears.

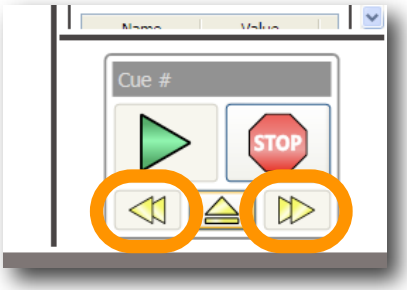


-or-

3. Press the **Load Cue** button on your Showstopper and select a cue from the dialog box that appears (just like #2).

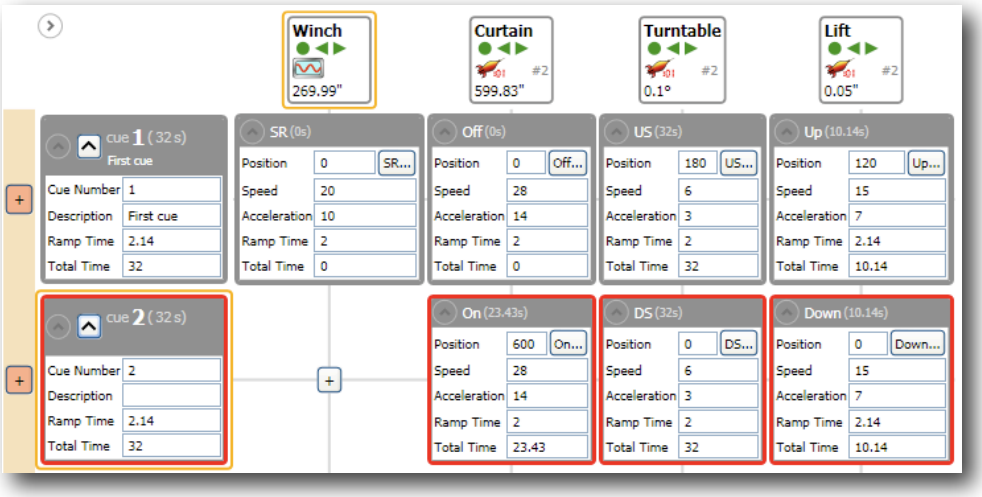
-or-

4. If a cue was loaded or run previously, you can use the **Next Cue** and **Previous Cue** buttons in the **Cue Controller** or on your Showstopper.

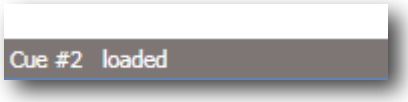


There are several visual indicators that let you know what movements are primed and ready.

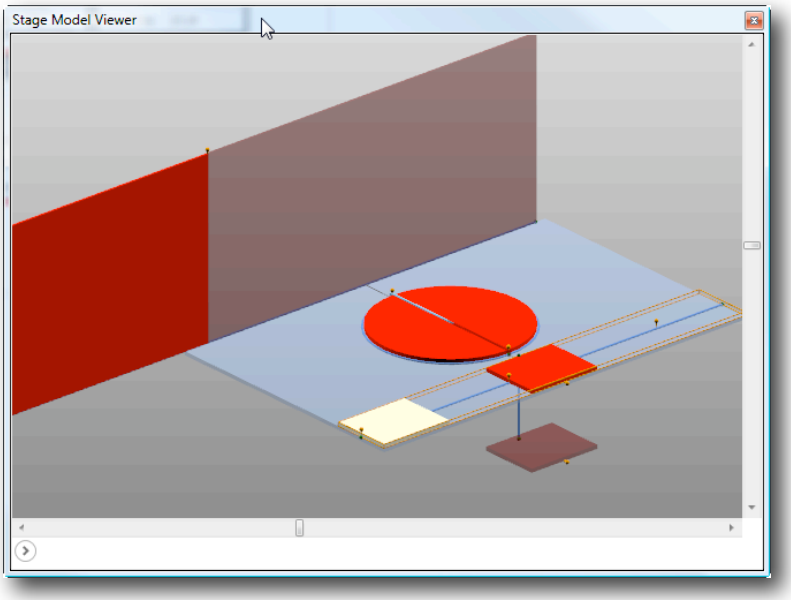
1. In the **Cue Grid**, the border around the cue and movements changes to red and the Stagehand column headers display a rocket ship countdown icon. *Note: If only some of the Stagehands were able to load the cue, the cue outline will turn pink indicating a partially loaded cue. This can occur if a Stagehand is not currently connected to SpikeMark.*



2. The status bar text updates in the bottom left corner.



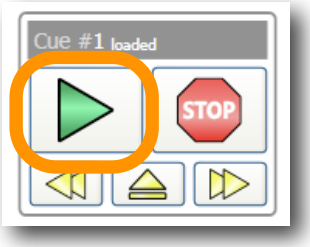
3. For motors that will move in the cue, the models inside the Stage Model Viewer change to red and their target position is shown in semi-transparent red.



Running a Cue

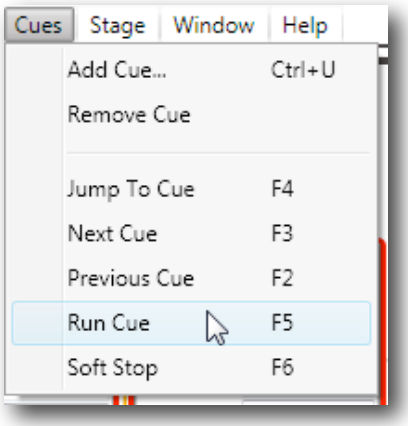
Once a cue is loaded, you can start all the motors moving by running the cue. There are a few ways to run a cue.

1. You can press the **GO** button in the **Cue Controller**.



-or-

2. From the menu bar select **Cues -> Run Cue**.



-or-

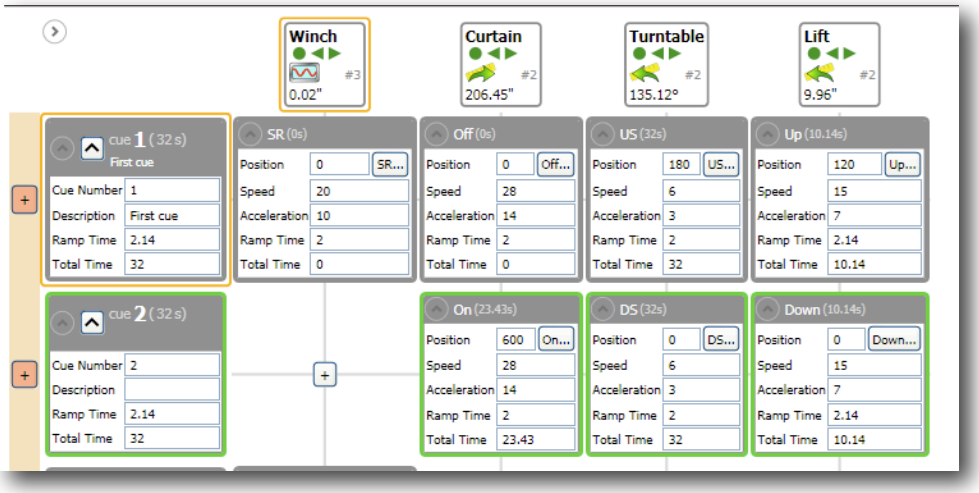
3. Press **F5** on the keyboard.

-or-

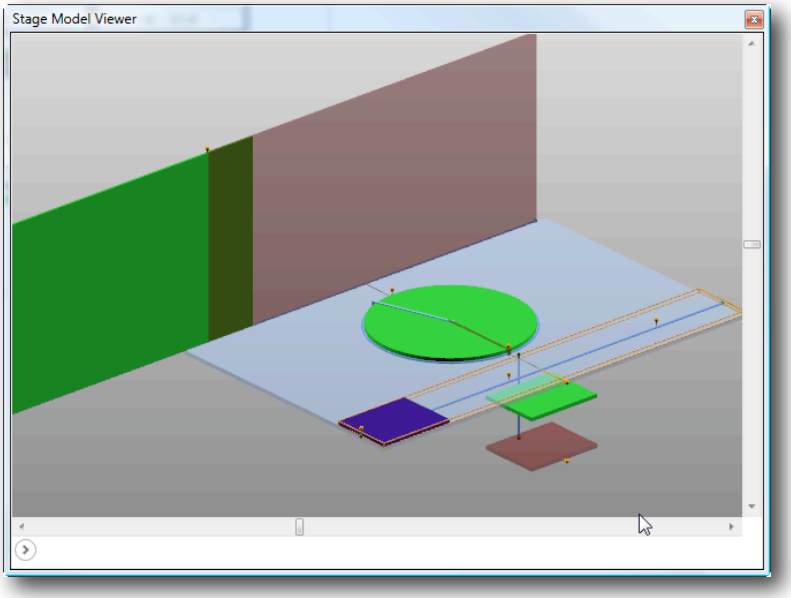
4. Press the **Go Cue** button on your Showstopper.

Once the cue is running, the interface changes to inform you what is running.

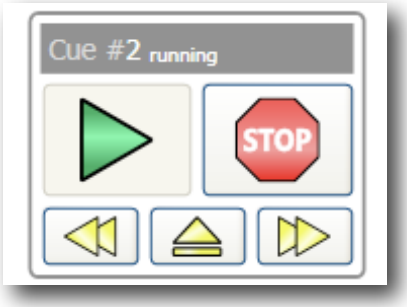
1. The **Cue Grid** borders change to green and the Stagehand column headers switch icons (see the sections on Motor Configuration and Stagehand FX Configuration for status icon descriptions).



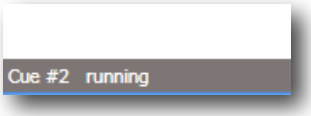
2. Motors in the **Stage Model** turn green.



3. The status text in the **Cue Controller** displays which cue is running. *Note: If you have multiple cues running at the same time, the Cue Controller only shows the last cue started.*

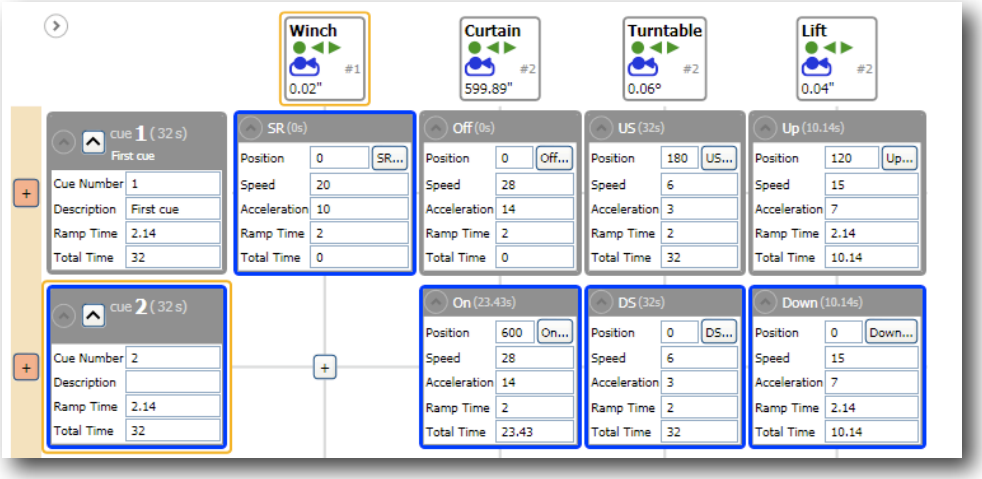


4. The status bar in the bottom left corner displays which cue is running. *Note: If you have multiple cues running at the same time, the Status Bar only shows the last cue started.*

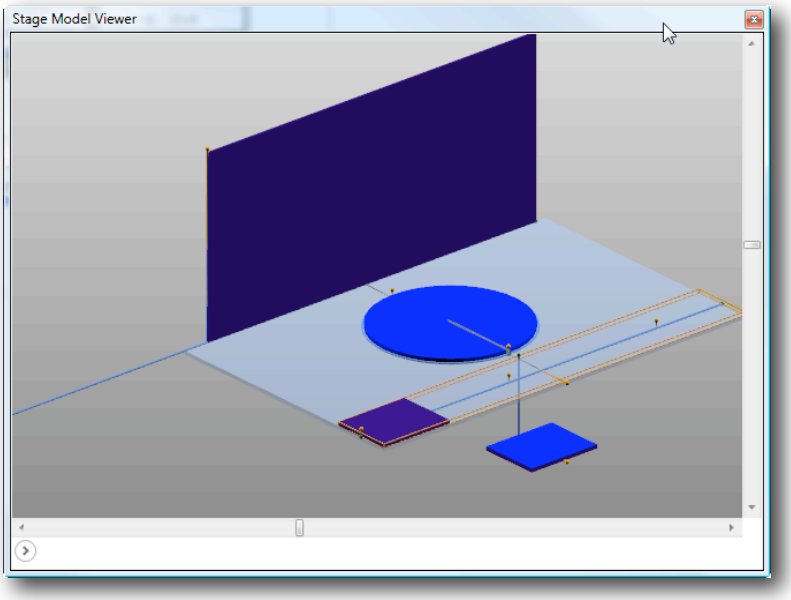


As the movements and cue complete, the interface shifts colors again.

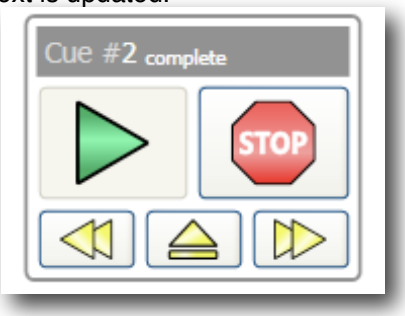
1. The borders in the **Cue Grid** turn blue.



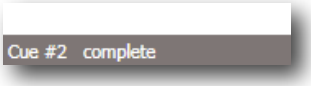
2. The **Stage Model** pieces turn blue.



3. The **Cue Controller** status text is updated.



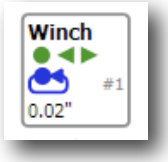
4. The **Status Bar** text is updated.



Deciphering Stagehand Status During the Show

In the heat of a performance, you are likely to feel compelled to obsessively check that your motors and fx's are all connected, happy, and ready to run. There's no shame in that compulsion, it's what makes you a good automation operator. If there is a problem, you want to know about it so that you can fix it quickly and be ready to run your next cue. SpikeMark feeds that hunger for status reports with a few key signals in the interface.

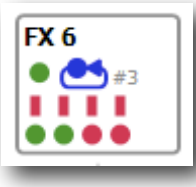
The **Cue Grid** header cells give you quick status information about every motor and fx. Let's take a look at a motor header first.



From left to right, top to bottom, the information in a motor header cell is:

1. The motor name.
2. The green circle indicates the network connection status. Green is connected, yellow is connecting, red is disconnected.
3. The left arrow is the reverse limit switch status. Green is clear, red is activated, gray is unknown (if the motor isn't connected on the network, SpikeMark can't tell you what state the limit switch is in).
4. The right arrow is the forward limit switch status, color coding is the same as the reverse limit switch.
5. The motor status icon shown is Complete, check the section *Motor Configuration* for a table of all the possible motor status icons.
6. The cue number is displayed to the right of the status icon. As shown in the above image, the motor has Completed Cue #1.
7. The motor position is the last bit of information. *If the motor is not connected currently to the network, this information may be stale!*

Let's examine the header cell for an fx.



Again, from left to right and top to bottom, the information in the fx header cell is:

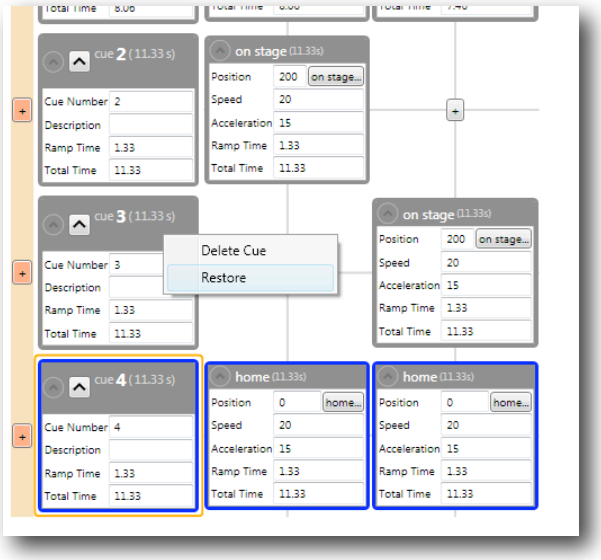
1. The name of the fx.

2. The green circle indicates the network connection status. Green is connected, yellow is connecting, red is disconnected.
3. The fx status icon shown is Complete, check the section *Stagehand FX Configuration* for a table of all the possible fx status icons.
4. The cue number is displayed to the right of the status icon. As shown in the above image, the fx has Completed Cue #3.
5. The row of rectangles represent the status of inputs 1 through 4 (the rectangle is supposed to look like an 'i' for input). Red means inactive, green means activated, gray means unknown (if the fx isn't connected to the network, SpikeMark has no clue what the status of the inputs are).
6. The row of circles represent the status of the outputs 1 through 4 (the circle is supposed to look like an 'o' for output). Red means inactive, green means activated, gray means unknown.

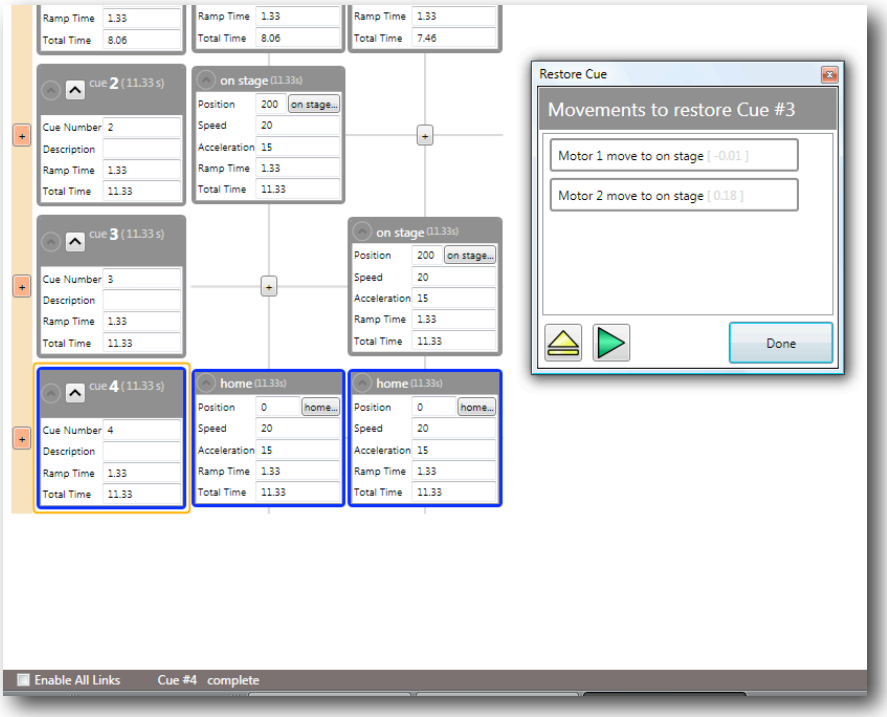
Restoring a Cue - *New in version 2.1!*

In SpikeMark v2.1, there's a handy new feature for moving backwards through the show during rehearsal. Cue Restore will put the every motor back into it's proper position for a specific cue, regardless of whether the motor was moving in that cue or had its last movement in a previous cue. If that sounds confusing, it is straightforward in practice. Let's take a look at how the feature works and you'll see that it does exactly what you need when jumping backwards in rehearsal.

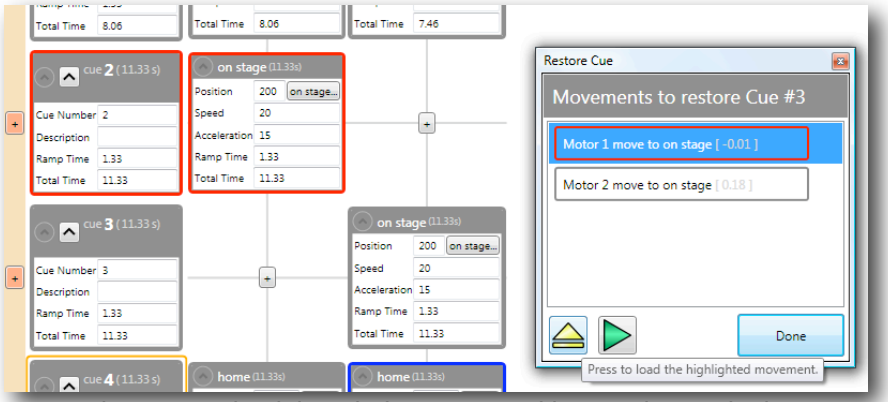
In the following example, let's assume that we just ran through cue #4 and the stage manager needs to re-run cue #4 because Jeff, the lighting designer, needs to look at the transition again. As you glance at the cue sequence below, you can see that we need to reset the stage so that we are sitting in cue #3. Ah, but there's a problem. You can't just run cue #3 to reset. If you were to just run cue #3, then Motor 1 would be left in it's "home" position. So you would need to run cue #3 & cue #2. Not a big deal in this scenario, but as you have more motors and more complex cues, restoring gets to be a real pain. Thankfully, Cue Restore can handle this easily. Let's see how it works.



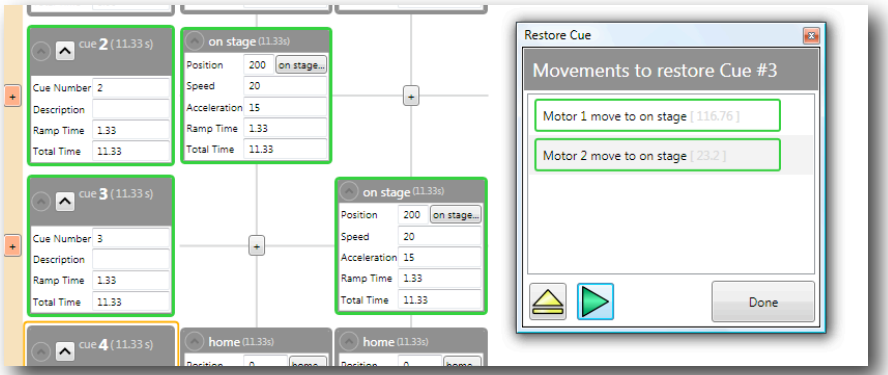
1. Right-click on Cue #3 and select **Restore**.


Restore Cue
Movements to restore Cue #3
Motor 1 move to on stage [-0.01]
Motor 2 move to on stage [0.18]
Done
Enable All Links
Cue #4 complete

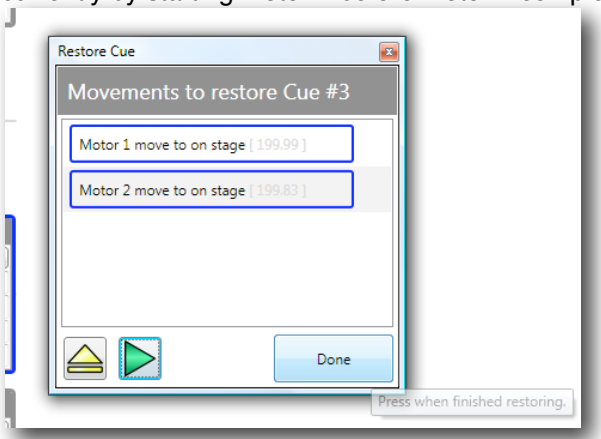
2. A list is presented showing that both Motor 1 & Motor 2 need to move in order to restore Cue #3. *Note: There's a new checkbox in the status bar at the bottom of the screen. "Enable All Links" can be used to turn off cue links. The Cue Restore feature automatically turns of all links so that you don't accidentally trigger a sequence during the restore.*


Restore Cue
Movements to restore Cue #3
Motor 1 move to on stage [-0.01]
Motor 2 move to on stage [0.18]
Done
Press to load the highlighted movement.

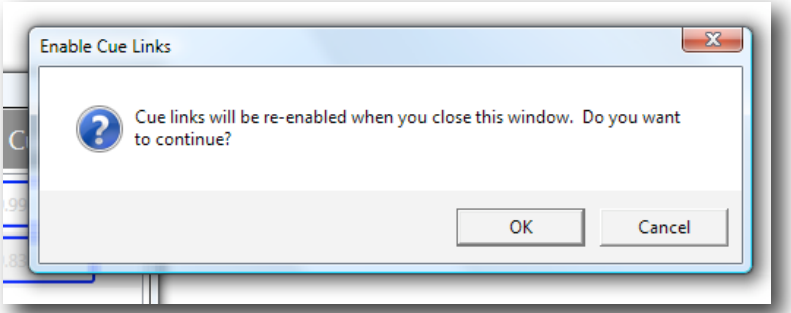
3. To start moving motors back into their correct positions, select a single movement from the list and then press the "Load Movement" button in the bottom of the Restore Cue window. Load and run movements one by one (though you don't have to wait for each one to complete before starting the next one). Why? Because SpikeMark has no way to know what order to restore the motors. You need to look at the stage and confirm that restoring a motor won't cause a collision on stage.



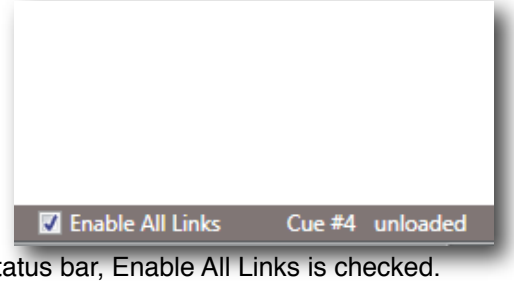
4. After one movement is loaded, you can press the “Run” button. Then, if it’s safe, go ahead and load the next movement and run it as well. Here you can see that we are running two movements concurrently by starting Motor 2 before Motor 1 completed.


Restore Cue
Movements to restore Cue #3
Motor 1 move to on stage [199.99]
Motor 2 move to on stage [199.83]
Done
Press when finished restoring.

5. Once you’re finished restoring all the motors, press the Done button. This closes the Restore Cue window and re-enables cue links.



6. A prompt is displayed to confirm that you are re-enabling cue links. If you motors haven’t finished moving yet, click Cancel.



7. Notice that in the status bar, Enable All Links is checked.

Now you're all ready to run Cue #4 again, and if Jeff still doesn't have the lighting right it's easy for you to restore again!

Troubleshooting

When automation works well, it's a fun and wondrous technology. When automation doesn't work it's hair-pulling, frustrating black-magic. 'Why the heck didn't the motor move!!!!' If you've made it to the part of the manual, I assume you are at the hair-pulling stage. I'll try to help answer some of the more common problems and solutions here. If these suggestions don't address your problem, give us a call, send an email, or use the discussion forums.

Contact Us

Phone 401-862-2980 (free tech support for the first 90 days, after that $30 / half-hour fees may apply at our discretion)

Email support@creativeconners.com (always free, response times vary from minutes to a couple of hours)

Web forums can be found at http://creativeconners.com/phpBB2

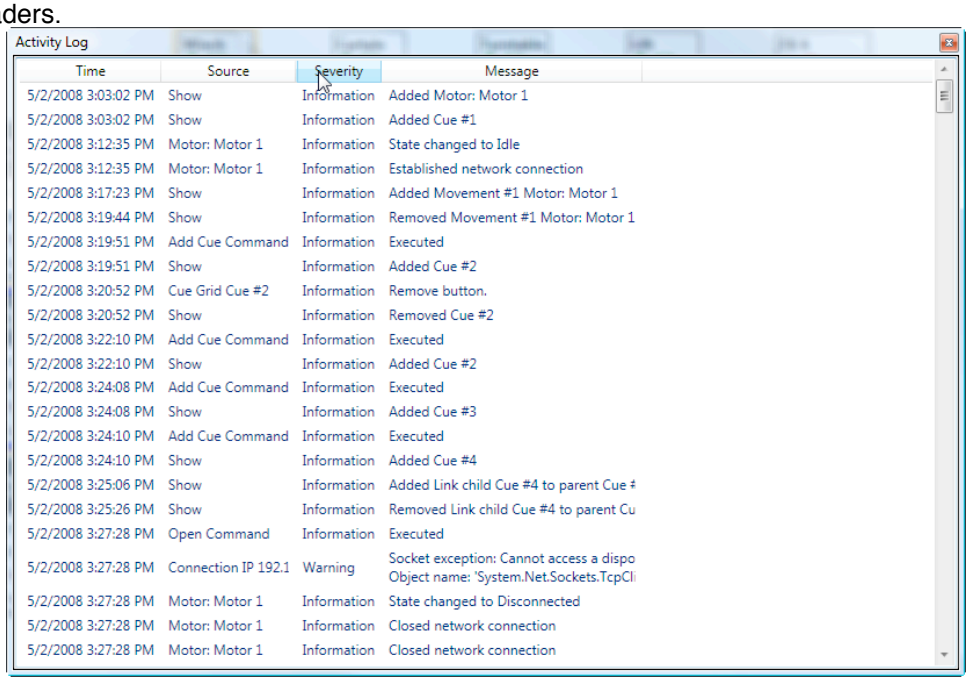
The Stagehand network connection doesn't work

1. The most common reason for this is that the IP address between the Stagehand and the computer are not compatible. Take a look at the Quick Start chapter of the manual under the *Setup a Network Connection* section.
2. Check your network cabling. Swap out the cables, preferably with a cable you know is good from another working Stagehand.
3. Try resetting the IP address of the Stagehand to a slightly different address, and then update the address in SpikeMark. If the network server in the Stagehand has stalled, resetting its IP address will get it started again.
4. Make sure that you have a network hub between the computer and the Stagehand. If you are trying to connect directly from the computer to the Stagehand you need a special kind of network cable known as a Crossover Cable.
5. Swap out the network hub.

A motor didn't start to move when I ran a cue

1. Check to see if the motor is sitting on a limit switch. Limit switch status is shown in the Cue Grid and the Stagehand Editor.
2. Make sure you have a Proportional Gain of at least 1 in the Tuning section of the Stagehand Editor.
3. Make sure that the cue had a speed and acceleration more than zero (0).
4. Check the activity log to get some information about what happened. Select Window -> Log Viewer from the menu bar. The Log Viewer is displayed and it shows all the actions that

SpikeMark is executing behind the scenes. You can sort the log by clicking on the column headers.


Activity Log
Time
Source
Severity
Message
5/2/2008 3:03:02 PM Show Information Added Motor: Motor 1
5/2/2008 3:03:02 PM Show Information Added Cue #1
5/2/2008 3:12:35 PM Motor: Motor 1 Information State changed to Idle
5/2/2008 3:12:35 PM Motor: Motor 1 Information Established network connection
5/2/2008 3:17:23 PM Show Information Added Movement #1 Motor: Motor 1
5/2/2008 3:19:44 PM Show Information Removed Movement #1 Motor: Motor 1
5/2/2008 3:19:51 PM Add Cue Command Information Executed
5/2/2008 3:19:51 PM Show Information Added Cue #2
5/2/2008 3:20:52 PM Cue Grid Cue #2 Information Remove button.
5/2/2008 3:20:52 PM Show Information Removed Cue #2
5/2/2008 3:22:10 PM Add Cue Command Information Executed
5/2/2008 3:22:10 PM Show Information Added Cue #2
5/2/2008 3:24:08 PM Add Cue Command Information Executed
5/2/2008 3:24:08 PM Show Information Added Cue #3
5/2/2008 3:24:10 PM Add Cue Command Information Executed
5/2/2008 3:24:10 PM Show Information Added Cue #4
5/2/2008 3:25:06 PM Show Information Added Link child Cue #4 to parent Cue
5/2/2008 3:25:26 PM Show Information Removed Link child Cue #4 to parent Cu
5/2/2008 3:27:28 PM Open Command Information Executed
5/2/2008 3:27:28 PM Connection IP 192.1 Warning Socket exception: Cannot access a dispo
Object name: 'System.Net.Sockets.TcpCl
5/2/2008 3:27:28 PM Motor: Motor 1 Information State changed to Disconnected
5/2/2008 3:27:28 PM Motor: Motor 1 Information Closed network connection
5/2/2008 3:27:28 PM Motor: Motor 1 Information Closed network connection

SpikeMark showed a strange error message then I had to quit

That's a bug, and we need to fix it. Please email a copy of your show file, a copy of the Activity Log, and a brief description of any pertinent information to support@creativeconners.com.

I can't find the Log file to email it to support

Log files are now stored in *C:\Documents and Settings\Current User Account\Application Data\Creative Conners*. The **Application Data** folder is hidden by default on some Windows installations. To view the folder, you must turn on an option from the menu bar in **Windows Explorer**. Select the menu **Tools -> Folder Options... -> View -> Show hidden files and folders**.

A motor starts to run in a cue but stops abruptly before getting to spike

The motor isn't tuned well enough to reach it's target position.

1. Tune the motor better if possible. See the section on motor tuning.
2. In the **Stagehand Editor** uncheck the **Strict Timing Mode** box in the **Cue Behavior** section. This will let the motor keep trying to get to target position even if the tuning is sub-optimal.
3. If the motor is showing a **Position Error** fault and you are certain that nothing is physically stuck in the machinery, you can raise the **Max Position Error** value in the **Position** section of the **Stagehand Editor**.
4. If the motor is showing a **Position Error** fault, you can turn off the **Abort On Position Error** option inside the **Position** section of the **Stagehand Editor**. This is not a great thing to do. If the encoder becomes broken or disconnected, nothing will stop the motor from running away. Since no feedback is being sent to the Stagehand, the Stagehand will send increasing amounts of power to the motor until it is at full speed and you will need to use the Emergency Stop to

shutdown the motor. See the section *Max Position Error* in the *Motor Configuration* chapter for more information.

The motor runs full speed in the wrong direction

The motor and encoder polarity are out of sync. Swap two of the power leads in the motor plug or swap the encoder signals A & /A with B & /B inside the encoder plug.

Appendix A - Basic Motion Control Concepts

From the perspective of an audience member motion control is magic. From the technician's perspective it should not be magic. Though the combination of Stagehand™ motor controllers and SpikeMark™ software create a motion control system that everyone can use, it is important to know some basic concepts to avoid mistakes and help troubleshoot your system.

Power

Depending on the model of Stagehand™ that you are using you need to supply either 120vac or 230vac power to each Stagehand™. The Stagehand™ with convert this power into something suitable for your motor. The output voltage, or frequency for AC motors, is varied to achieve variable motor speed.

The Stagehand™ tracks the motor's position and then relays that position to SpikeMark™ to display on-screen. If a Stagehand™ loses power, the next time it powers up it will reset its position to zero (0). This is very important, because if a motor is sitting 10' from centerline and the power is interrupted, when the power comes back on the Stagehand™ will think that the motor is now at 0' from centerline. In this case, the motor needs to be driven back to the true 0' spike (usually a piece of spike tape or limit switch) and reset to 0', either through SpikeMark™ (using the **Reset Position** button on the **Stagehand Editing Pane**) or by cycling power to the Stagehand™.

The best operating procedure is to return all motors to zero (0) at the end of every performance. If power is lost overnight when no one is around, then it won't affect the Stagehand™'s position information. A pre-show check should be done every night to insure that every motor is truly at the correct position.

Since all position information is stored in the Stagehand™, the PC running SpikeMark™ can be shut down at any time without affecting position information whatsoever.

Position Feedback

How does the Stagehand™ know how to change speeds? Good question. There are two factors that affect the speed of a motor. The first is the target speed that is programmed for each cue. In SpikeMark™ you tell each motor how fast to move in each cue. This information is sent to the appropriate Stagehand™, which in turn supplies some voltage to the motor and watches to see what happens. If the motor moves faster than the Stagehand™ expected, it reduces the voltage. If the motor

moves slower than expected, the voltage is raised. This cycle continues at a rapid pace until the motor reaches the desired position.

In order for the Stagehand™ to know how far and how fast the motor is moving in relation to the voltage it is supplying it needs an encoder. An encoder is a simple device that creates electronic pulses as it spins. The encoder is physically connected to the motor so that as the motor spins the encoder spins. If the encoder were disconnected from the motor, the Stagehand™ would not see any movement regardless of whether the motor was actually rotating or not. This could cause the motor to race, because the Stagehand™ would keep supplying more and more voltage to the motor hoping to see an increase in speed. A similar situation would occur if the polarity of the motor was reversed from the encoder.

There are safety features that you can set up to prevent a motor from racing, such as **Max Position Error** which tells a Stagehand to shutdown if it doesn't get good response from the motor. This feature is discussed in the chapter *Motor Configuration*. Later in this section *Over-travel Limits* are discussed as another safety precaution to guard against motor racing.

Motor Tuning

The cycle of watching encoder pulses and adjusting motor voltage is controlled by the tuning parameters of the Stagehand™. These tuning parameters are often referred to as the PID filter, which stands for Proportional, Integral, and Derivative gains. You enter numeric values for this filter into SpikeMark™, which will then send the filter parameters to a Stagehand™. While the mathematics of this filter is somewhat complex, the practical application is not. We walked through motor tuning in greater detail in the chapter *Stagehand Configuration*.

Over-travel Limits

If the encoder position is wrong because of loss of power, physical disconnect from the motor, or some equipment failure, a physical limit switch can be used to protect the motor from running too far. These limit switches, when struck, will prevent the motor from moving any farther in the detected direction (either forward or reverse). The Stagehand™ will only permit a movement in the opposite direction until the limit switch is cleared.

Those are the basic principles of motion control, as you get more familiar with the operation of SpikeMark™ and Stagehand™ these concepts with become more concrete and intuitive.